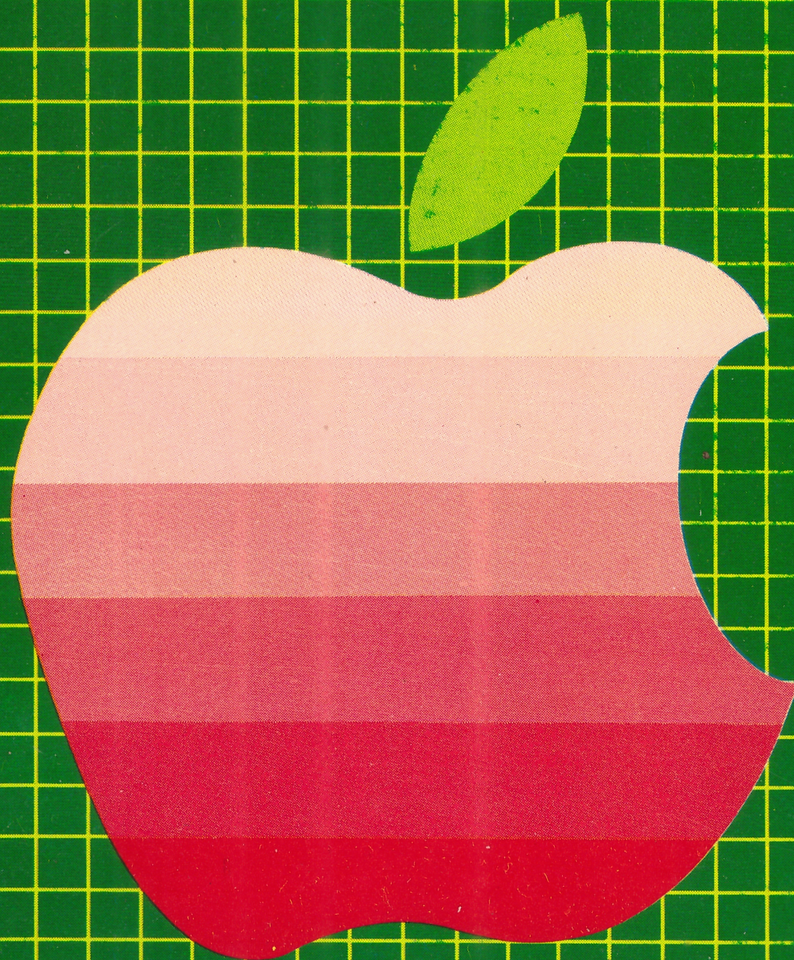
เรียน APPLESOFT BASIC ด้วยตนเอง เล่ม 2

ประพัฒน์ อุทโยภาศ



ซีเอ็ด

เรียน APPLESOFT BASIC ด้วยตนเอง เล่ม 2

ประพัฒน์ อุทโยภาศ

ROBINSON
25-4-29
20.20.



บริษัท ซีเอ็ดยูเคชั่น จำกัด
SCIENCE, ENGINEERING & EDUCATION CO.,LTD.

ราคา 42 บาท

ISBN 974-7965-03-8

พิมพ์ครั้งที่ 1 พ.ศ. 2526
พิมพ์ครั้งที่ 2 พ.ศ. 2526
พิมพ์ครั้งที่ 3 พ.ศ. 2527
พิมพ์ครั้งที่ 4 พ.ศ. 2527
พิมพ์ครั้งที่ 5 พ.ศ. 2528

จัดพิมพ์และจัดจำหน่ายโดย



บริษัท ซีเอ็ดยูเคชั่น จำกัด
SCIENCE, ENGINEERING & EDUCATION CO.,LTD.

800/43-45 ซอยตระกูลสุข ถนนอโศกดินแดง
เขตห้วยขวาง กท. 10400 โทร.245-0390

พิมพ์ที่ หจก. เอช-เอน การพิมพ์
402/16 ซอยดอนกุศล เขตยานนาวา กทม. 10120
นายประพันธ์ จิระมงคล ผู้พิมพ์ผู้โฆษณา พ.ศ. 2528

คำนำ

เมื่อหนังสือเล่มแรกของผู้เขียนชื่อ "เรียน APPLESOFT BASIC ด้วยตนเอง" ออกวางตลาด และได้ทราบจากสำนักพิมพ์ว่าหนังสือจำหน่ายได้หมดภายในเวลา 20 วันจะต้องพิมพ์ครั้งที่ 2 แต่ต่อมาอีกไม่นานก็มีการพิมพ์ครั้งที่ 3 ผู้เขียนเกิดความปิติยินดีอย่างยิ่ง ความปิติยินดีมิใช่เกิดเนื่องจากผู้เขียนหวังว่าจะได้เงินเป็นค่าลิขสิทธิ์เป็นจำนวนมากขึ้นแต่อย่างเดียว หากใจจริงของผู้เขียนยินดีที่ได้ทราบโดยมิต้องมีการสำรวจวิจัยแต่อย่างใดว่าปัจจุบันนี้ในบ้านเราได้มีการตื่นตัวในเรื่องการใช้ไมโครคอมพิวเตอร์กันอย่างมาก และนี่ก็คือนิมิตอันดีที่บ่งชัดว่าความเจริญในด้านวิทยาการและการธุรกิจคงจะเป็นไปอย่างรวดเร็วเพราะไมโครคอมพิวเตอร์คือ อุปกรณ์ที่จะช่วยเหลือในเรื่องนี้ได้อย่างมหาศาล

ผู้เขียนได้ทราบว่ามีโรงเรียนสอนไมโครคอมพิวเตอร์บางโรงเรียนประสบความสำเร็จในการที่นำหนังสือของผู้เขียนไปใช้ประกอบการสอน ด้วยเหตุผลที่ว่า "เนื้อหาก็ไม่เท่าไรเหมือนหนังสืออื่น แต่สามารถทำให้เรื่องโปรแกรมคอมพิวเตอร์เป็นเรื่องง่าย คนที่ไม่รู้อะไรเลยก็สามารถเรียนรู้ได้" นี่ก็นับว่าผู้เขียนได้บรรลุถึงวัตถุประสงค์โดยสิ้นเชิงแล้ว

ทั้งหมดที่กล่าวมานี้ เป็นแรงบันดาลใจให้ผู้เขียนมีความมานะเขียนเล่มต่อไปอีก คือ เล่มที่ท่านกำลังอ่านอยู่นี้ โดยนำเอาสิ่งที่ผู้เขียนละไว้ในเล่มก่อน เพราะ

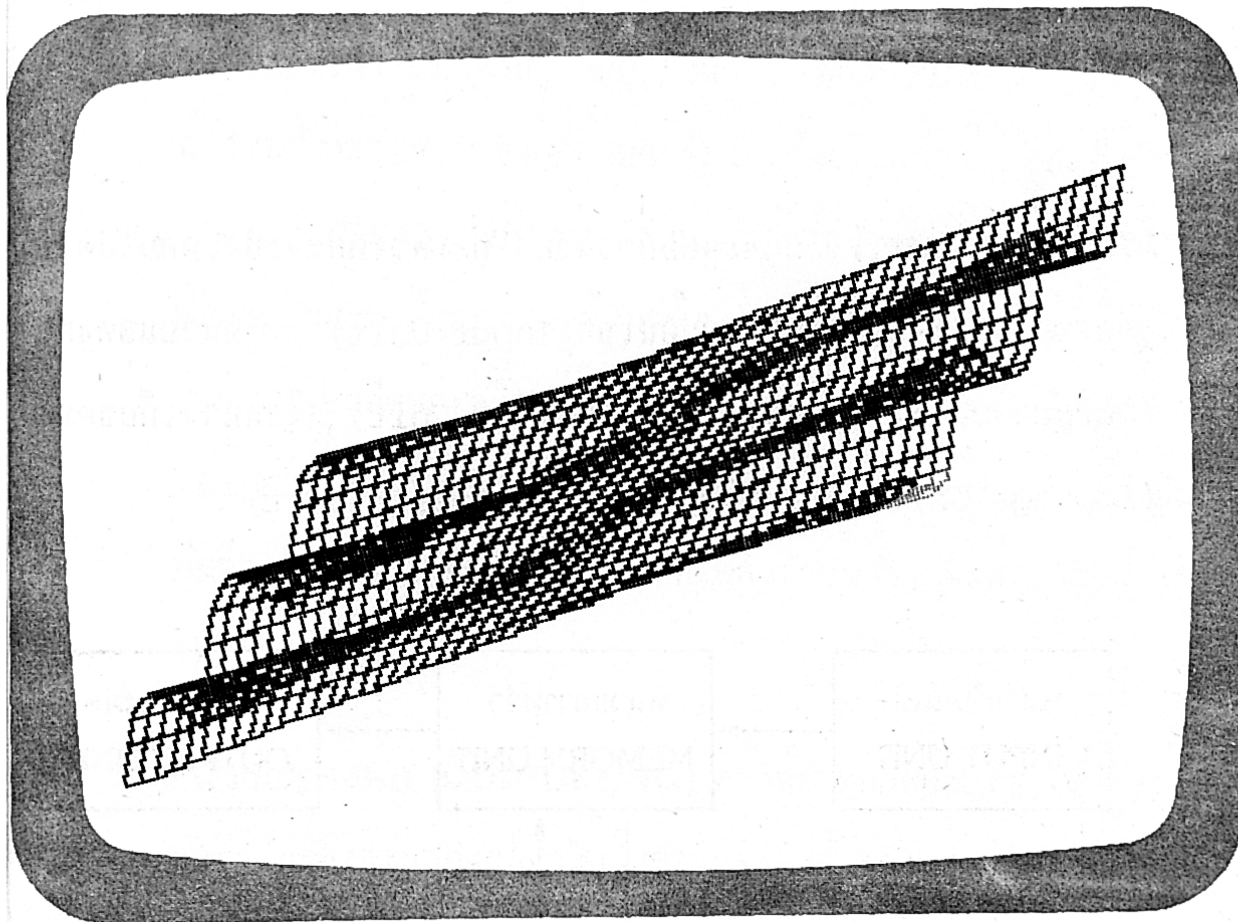
เห็นว่าอาจจะทำให้สับสนสำหรับผู้เริ่มแรกมาบรรจุไว้ในเล่มนี้ พร้อมด้วยเรื่องอื่นที่ควรจะรู้ในการศึกษาขั้นต่อไป ข้อจำกัดในการใช้หนังสือเล่มนี้มีอยู่ว่าท่านควรจะศึกษาหนังสือเล่มแรกอย่างละเอียดเสียก่อนแล้วจึงจะเริ่มอ่านเล่มนี้และท่านจำเป็นจะต้องมีไมโครคอมพิวเตอร์อยู่ใกล้มือพอที่จะทำการทดลองโปรแกรมได้

ผู้เขียนหวังเป็นอย่างยิ่งว่า ท่านผู้อ่านคงจะประสบความสำเร็จในการทำความเข้าใจ และรู้จักใช้โปรแกรมให้เป็นประโยชน์ไม่ว่าในทางบันเทิง ในธุรกิจหรือในการศึกษา

ประพัฒน์ อุทโยภาศ

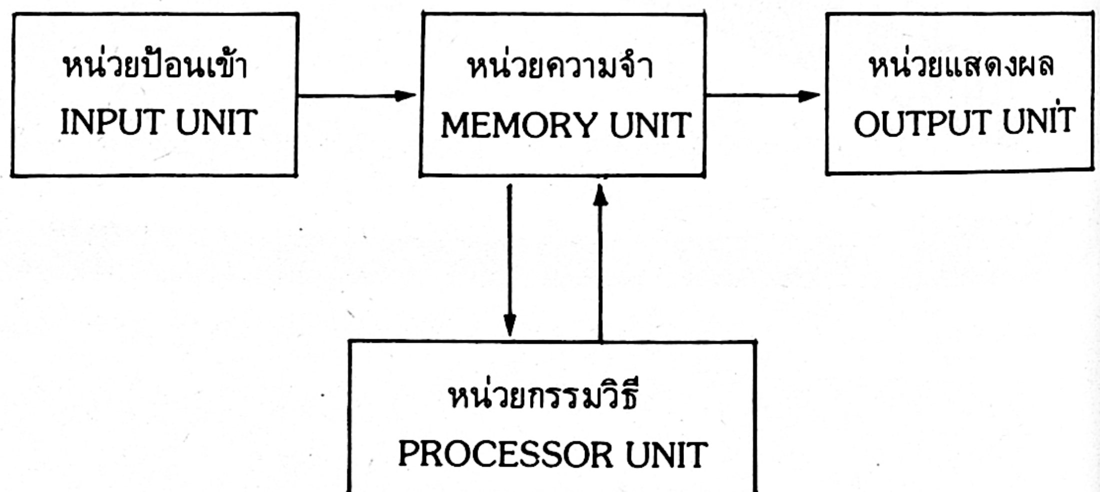
สารบัญ

บทที่ 1 ลึกเข้าไปในคอมพิวเตอร์ 8
บทที่ 2 เพิ่มเติมเสริมแต่ง 20
บทที่ 3 FUNCTIONS 34
บทที่ 4 กราฟิก 48
บทที่ 5 SHAPE TABLE 70
บทที่ 6 TEXT FILE 88
บทที่ 7 FLOW CHART 128
ภาคผนวก 137
หนังสืออ้างอิง 138
ดัชนี 139



1 ลึกเข้าไปในคอมพิวเตอร์

1.1 **ระบบคอมพิวเตอร์** ถ้าจะพูดอย่างง่ายๆ คอมพิวเตอร์จะประกอบไปด้วยอุปกรณ์ต่างๆ ดังนี้ คือ หน่วยป้อนเข้า(Input Unit) หน่วยแสดงผล (Output Unit) หน่วยความจำ(Memory Unit) หรือหน่วยเก็บข้อมูล (Storage Unit) และหน่วยกรรมวิธี(Processor Unit)



รูปที่ 1.1 ระบบคอมพิวเตอร์

อุปกรณ์ต่างๆ เหล่านี้ ภาษาคอมพิวเตอร์เรียกว่า *HARDWARE* ซึ่งก็คือส่วนประกอบทั้งหลายที่สามารถมองเห็นได้และสัมผัสได้ เช่น แป้นพิมพ์ เครื่องพิมพ์ เครื่องบันทึก จอภาพ เป็นต้น *หน่วยป้อนเข้า*ก็คือ ทางที่คนจะบอก

เครื่อง *หน่วยแสดงผล*ก็คือ ทางที่เครื่องจะบอกคน *หน่วยความจำ* ก็คือ สมองส่วนที่ใช้จำ และ*หน่วยกรรมวิธี* ก็คือ สมองส่วนที่ใช้คิด

1.2 **หน่วยป้อนเข้า (INPUT UNIT)** คือเครื่องมือที่จะติดต่อให้เครื่องทำงานให้ อาจจะโดยการชี้แนะให้ว่าจะทำอย่างไร โดยใช้ข้อมูลอะไร

หน่วยป้อนเข้าจะมีหลายรูปแบบ เช่น แป้นพิมพ์(KEYBOARD) ซึ่งปกติจะติดตั้งรวมกับหน่วยความจำ-หน่วยกรรมวิธี(ซึ่งเรียกว่าCentral Processor Unit - CPU) แป้นพิมพ์จะติดต่อกับเครื่องโดยตรง ด้วยการที่คนกดแป้นป้อนโปรแกรม ข้อมูล หรือคำสั่งเข้าไป เครื่องอ่าน-บันทึกจาน(DISK DRIVE) จะติดต่อเครื่องโดยการอ่านข้อมูลโปรแกรมหรือคำสั่งจากแผ่นจาน (DISKETTE หรือ DISK) แล้วส่งเข้าเครื่อง ถ้าหน่วยป้อนเข้าเป็นเครื่องบันทึกเสียงแบบเทป(แบบที่ใช้บันทึกเพลงธรรมดา) จะส่งข้อมูลต่างๆ จากเทปเข้าเครื่อง

1.3 **หน่วยแสดงผล (OUTPUT UNIT)** คล้ายกับหน่วยป้อนเข้าเพียงแต่กลับทางกัน เครื่องจะแสดงผลมาในรูปของผลของการคำนวณ บอกข้อผิดพลาดหรือถามว่าจะทำอย่างไรต่อไป หน่วยแสดงผลทั่วไปอยู่ในรูปของจอภาพ(MONITOR หรือCATHODE RAY TUBE-CRT) และเครื่องพิมพ์(PRINTER)

1.4 **หน่วยความจำ** หน่วยความจำมี 2 แบบคือ หน่วยความจำภายใน(Internal Memory) และหน่วยความจำสำรอง(Mass Memory)

1.4.1 หน่วยความจำภายในจะรวมอยู่ใน Central Processor Unit หรือ CPU ซึ่งเปรียบเสมือนตู้เก็บของที่เป็นช่องๆ แต่ละช่องก็มีหมายเลขของตัวเองอยู่และสามารถเก็บเอกสารได้ช่องละชิ้น หน่วยความ

จำภายในยังแบ่งออกได้เป็น 2 ชนิดคือ RAM ซึ่งย่อมาจาก Random Access Memory เป็นช่องที่เราสามารถสั่งให้คอมพิวเตอร์บรรจุข่าวสารลงไปได้และสั่งให้เอามาใช้ได้ ส่วนช่องที่เรียกว่า ROM (Read Only Memory) เป็นช่องที่ผู้สร้างได้บรรจุข่าวสารเข้าไปแล้ว เราสามารถจะเอาข่าวสารมาใช้ได้แต่จะบรรจุเข้าไปไม่ได้

เหตุผลที่คอมพิวเตอร์ต้องมีทั้ง RAM และ ROM ก็มีอยู่ว่าถ้าหากคอมพิวเตอร์มีแต่ RAM อย่างเดียว เมื่อปิดกระแสไฟที่เข้าเครื่อง ความจำที่มีอยู่ใน RAM จะหายไปหมด คอมพิวเตอร์ก็จะปฏิบัติงานไม่ได้ เพราะไม่มีหน่วยความจำที่จะสั่งให้ทำงานนอกจากจะป้อนเข้าไปใหม่ ถ้าหากมีแต่ ROM อย่างเดียว เมื่อมีการป้อนข้อมูลเข้า ข้อมูลที่มีอยู่เดิม ซึ่งเป็นข้อมูลที่สั่งให้เครื่องทำงานจะหายไป เครื่องก็จะปฏิบัติงานไม่ได้เช่นกัน ด้วยเหตุนี้จึงต้องมีทั้ง RAM และ ROM สรุปแล้ว ROM คือ หน่วยความจำที่เครื่องใช้ ส่วน RAM คือ หน่วยความจำที่ผู้ใช้เครื่องจะใช้ได้



รูปที่ 1.2 E.T. มี 8 นิ้ว

คอมพิวเตอร์จำได้อย่างไร นี่คือปัญหาที่ทุกคนอยากทราบ ขอย้อนกล่าวไปถึงเมื่อมนุษย์ยังนับไม่เป็น เมื่อคิดจะนับขึ้นมาก็ต้องหาว่าควรจะจัดหมวดหมู่อย่างไร เมื่อมนุษย์มีสิบนิ้วก็คิดว่าควรจะจัดหมู่ละสิบ คือนับตั้งแต่หนึ่งไปถึงสิบ แล้วเริ่มต้นใหม่ที่สิบหนึ่ง(สิบเอ็ด) ไปจนถึงสองสิบ(ยี่สิบ) เลขที่มีหมวดหมู่ดังกล่าวเรียกว่า *เลขฐานสิบ*

E.T. ลงมาจากดาวที่ไกลโพ้น E.T. มี 8 นิ้ว ก็ย่อมเป็นที่แน่นอนว่าบนดวงดาวที่เป็นแหล่งกำเนิดของ E.T. ต้องใช้เลขฐานแปดแทนที่จะเป็นฐานสิบแบบโลกมนุษย์

ย้อนมาพูดถึงคอมพิวเตอร์บ้าง คอมพิวเตอร์ก็คือเครื่องใช้ไฟฟ้าธรรมดาที่ต้องใช้กระแสไฟฟ้าในการทำงาน คอมพิวเตอร์ไม่มีนิ้ว(มีแต่แป้นพิมพ์ให้คน

A	• —	1	• — — — —
B	— • • •	2	• • — — —
C	— • — •	3	• • • — —
D	— • •	4	• • • • —
E	• (1 unit)	5	• • • • •
F	• • — •	6	— • • • •
G	— — •	7	— — • • •
H	• • • •	8	— — — • •
I	• •	9	— — — — •
J	• — — —	0	— — — — —
K	— • —		
L	• — • •	period	• — • — • —
M	— —	comma	— — • • — —
N	— •	colon	— — — • • •
O	— — —	query	• • — — • •
P	• — — •	apostrophe	• — — — — •
Q	— — • —	hyphen	— • • • • —
R	• — •	fraction bar	— • • — •
S	• • •	parentheses	— • — — • —
T	— (3 units)	quotation marks	• — • • — •
U	• • —		
V	• • • —		
W	• — —		
X	— • • —		
Y	— • — —		
Z	— — • •		

รูปที่ 1.3 รหัสมอร์ส

เราเอานิ้วกดเล่น) ถ้าคอมพิวเตอร์อยากจะนับบ้างก็ต้องอาศัยการเปิดปิดไฟฟ้าเป็นฐานของการนับ เอาเป็นว่าเปิดที่หนึ่ง(หรือประจุไฟฟ้าเป็นบวก) นับเป็น 1 ปิดที่หนึ่ง(หรือประจุไฟฟ้าเป็นลบ) นับเป็น 0 แต่นับแค่นี้จะเอาไปทำอะไรได้

เมื่อปี พ.ศ. 2381 ช่างเขียนภาพชาวอเมริกันคือ แซมมวล มอร์ส ได้คิดวิธีสื่อสารทางสาย ซึ่งปัจจุบันเรียกว่า *โทรเลข* โดยการส่งสัญญาณเป็นขีดและจุด เมื่อเอาขีดและจุดเหล่านี้มาเรียงสลับกันในลักษณะต่างๆ กัน ก็สามารถถอดออกมาเป็นตัวอักษรได้ สัญญาณดังกล่าวเรียกว่า *รหัสมอส*

กลับมาพูดถึงคอมพิวเตอร์อีก เมื่อมอร์สเอาเครื่องหมายขีดและจุดมาเรียงกันแล้วสามารถทำให้รู้เรื่องกันได้ ก็น่าจะเอาสัญญาณเปิดปิดมาเรียงกันได้เช่นกัน ดังนั้นผู้ประดิษฐ์คอมพิวเตอร์จึงได้ประดิษฐ์ให้ส่วนที่เล็กที่สุดของหน่วยความจำมีชื่อเรียกว่า BIT (ย่อมาจาก BInary digiT - เลขฐานสอง) ทำหน้าที่เก็บข่าวสารไว้ เพียงแต่ว่าเป็น 0 หรือ 1 เท่านั้นเอง ถ้าเอา 0 และ 1 อย่างละตัวมาเรียงกันจะได้เพียง 3 แบบคือ 01, 11 และ 10 เท่านั้น แต่ถ้าเอามาเรียงกัน 8 ตัว(ไม่กำหนดว่าอย่างละกี่ตัว) โดยวางสลับกันไปสลับกันมาไม่ให้ซ้ำกัน เราจะได้ถึง 2^8 หรือ 256 แบบ ซึ่งเราจะเอาไปใช้แทนตัวอักษรหรือตัวเลขได้อย่างเหลือเฟือทีเดียว

ลองดูว่าตัวเลขที่เราใช้กันอยู่เป็นฐานสิบ จะแทนด้วยเครื่องหมายที่เป็นฐานสองได้อย่างไร ดูรูปที่ 1.4

เลข 0 หรือ 1 จำนวน 1 ตัว เรียกว่า Bit (Binary Digit)
เลข 0 หรือ 1 จำนวน 8 ตัวมารวมกันเป็นชุด เรียกว่า Byte

ตัวเลขหรืออักษร	แทนด้วย
1	0 0 0 0 0 0 0 1
2	0 0 0 0 0 0 1 0
3	0 0 0 0 0 0 1 1
4	0 0 0 0 0 1 0 0
5	0 0 0 0 0 1 0 1
6	0 0 0 0 0 1 1 0
7	0 0 0 0 0 1 1 1
8	0 0 0 0 1 0 0 0
9	0 0 0 0 1 0 0 1
10	0 0 0 0 1 0 1 0
20	0 0 0 1 0 1 0 0
100	0 1 1 0 0 1 0 0
อักษร T	1 1 0 0 0 1 0 1
ฯลฯ	↓
	256 แบบ

รูปที่ 1.4 เลขฐานสอง (BINARY DIGIT)

Byte จำนวน 2^{10} ตัวหรือ 1024 ตัว เรียกว่า 1 Kilobyte หรือ K Byte หรือ KB

ตัวอักษร 1 ตัว ใช้ 1 BYTE

ตัวเลข 1 ตัว ใช้ 2-5 BYTE (เพราะต้องเผื่อเครื่องหมายและทศนิยม)

ดังที่กล่าวกันว่าคอมพิวเตอร์เครื่องนี้มีความสามารถ 48 K BYTE หมายถึงว่าหากจะป้อนตัวหนังสือเข้าไปคอมพิวเตอร์เครื่องนี้แล้ว RAM ทั้งหมดสามารถจะรับตัวอักษรได้ 48 × 1024 หรือ 49,152 ตัว หรือประมาณ 30 หน้ากระดาษจึงจะเต็ม(เมื่อเต็มแล้ว ก็สามารถถ่ายเข้าไปเก็บไว้ในหน่วยความจำสำรอง(Mass Memory) ได้ และสามารถรับได้ใหม่อีก ซึ่งจะได้กล่าวต่อไป)

1.4.2 **หน่วยความจำสำรอง** หน่วยความจำภายในของคอม-พิวเตอร์จะมีอยู่จำกัด ดังนั้นจึงจำเป็นต้องมีหน่วยความจำสำรอง ซึ่งจะมีอยู่ได้อย่างไม่จำกัด หน่วยความจำสำรองนี้เปรียบเสมือนที่พักของข้อมูลข่าวสาร หรือข้อชี้แนะ(Instruction) ซึ่งยังไม่ได้ใช้ในทันทีแต่อาจจะเรียกนำมาใช้ได้ หน่วยความจำภายในนั้นจะถูกเก็บไว้ด้วยประจุไฟฟ้า หากปิดสวิตช์ความจำก็จะสูญหายไปหมด แต่หน่วยความจำสำรองเก็บรักษาข้อมูลและข่าวสารไว้ด้วยสารแม่เหล็กเช่น เทปแม่เหล็กแบบที่ใช้อัดเสียงธรรมดานั่นเอง

หน่วยความจำในไมโครคอมพิวเตอร์เครื่องหนึ่งส่วนมากจะมีประมาณ 48 ถึง 64 K byte แต่ในหน่วยความจำสำรองเพียงหน่วยเพียงมีความจุเป็น 100 K byte และอาจจะถึง 5 หรือ 10 ล้าน Byte ได้

หน่วยความจำสำรองที่ใช้มี 3 แบบด้วยกัน แบบแรกคือ เทปบันทึกเสียงดังที่ได้กล่าวมาแล้ว ที่ดีกว่านั้นเป็นแบบ FLOPPY DISK (หรือ DISKETTE เป็นแผ่นคล้ายๆ จานเสียง มีเส้นผ่านศูนย์กลาง 2 ขนาดคือ 5¼ นิ้วและ 8 นิ้ว ที่ดีที่สุดคือ HARD DISK เป็นพลาสติกแข็ง รูปที่ 1.5 จะเป็นการเปรียบเทียบหน่วยความจำทั้ง 3 แบบนี้

	HARD DISK	FLOPPY DISK	เทปบันทึกเสียง
อุปกรณ์ประกอบ ลักษณะ	เครื่องอ่าน-บันทึกแผ่นจาน จานพลาสติกแข็ง ขนาด 12"	เครื่องอ่าน-บันทึกแผ่นจาน จานพลาสติกอ่อน ขนาด 5¼" หรือ 8"	เครื่องบันทึกเสียง เทปบันทึกเสียง
ความจุ	5 ล้าน Byte ขึ้นไป	80 KByte ถึง 1 ล้าน Byte	ไม่มาก
ความเร็ว	เร็วมาก	ปานกลาง	ช้ามาก
ราคาอุปกรณ์	1 แสนบาทขึ้นไป	10,000-20,000 บาท	800 บาท ก็ใช้ได้
ราคาแผ่น/เทป	รวมอยู่ในอุปกรณ์	100-150 บาท	15 บาท

รูปที่ 1.5 การเปรียบเทียบหน่วยความจำสำรอง

หน่วยความจำสำรองทั้ง 3 แบบนี้อาจจะใช้ทดแทนหรือสับเปลี่ยนกันได้ เช่น คอมพิวเตอร์ที่ใช้ HARD DISK อยู่ อาจจะถ่ายข้อมูล ข่าวสารลงไว้ใน FLOPPY DISK หลายๆ แผ่น เก็บไว้เผื่อในกรณีที่ข้อมูลใน HARD DISK ลบไป ปกติไมโครคอมพิวเตอร์จะใช้ FLOPPY DISK หากผู้เล่นไม่สามารถจะซื้อเครื่องอ่าน-บันทึกได้ก็อาจจะใช้เครื่องบันทึกเสียงธรรมดาไปพลางก่อนก็ได้

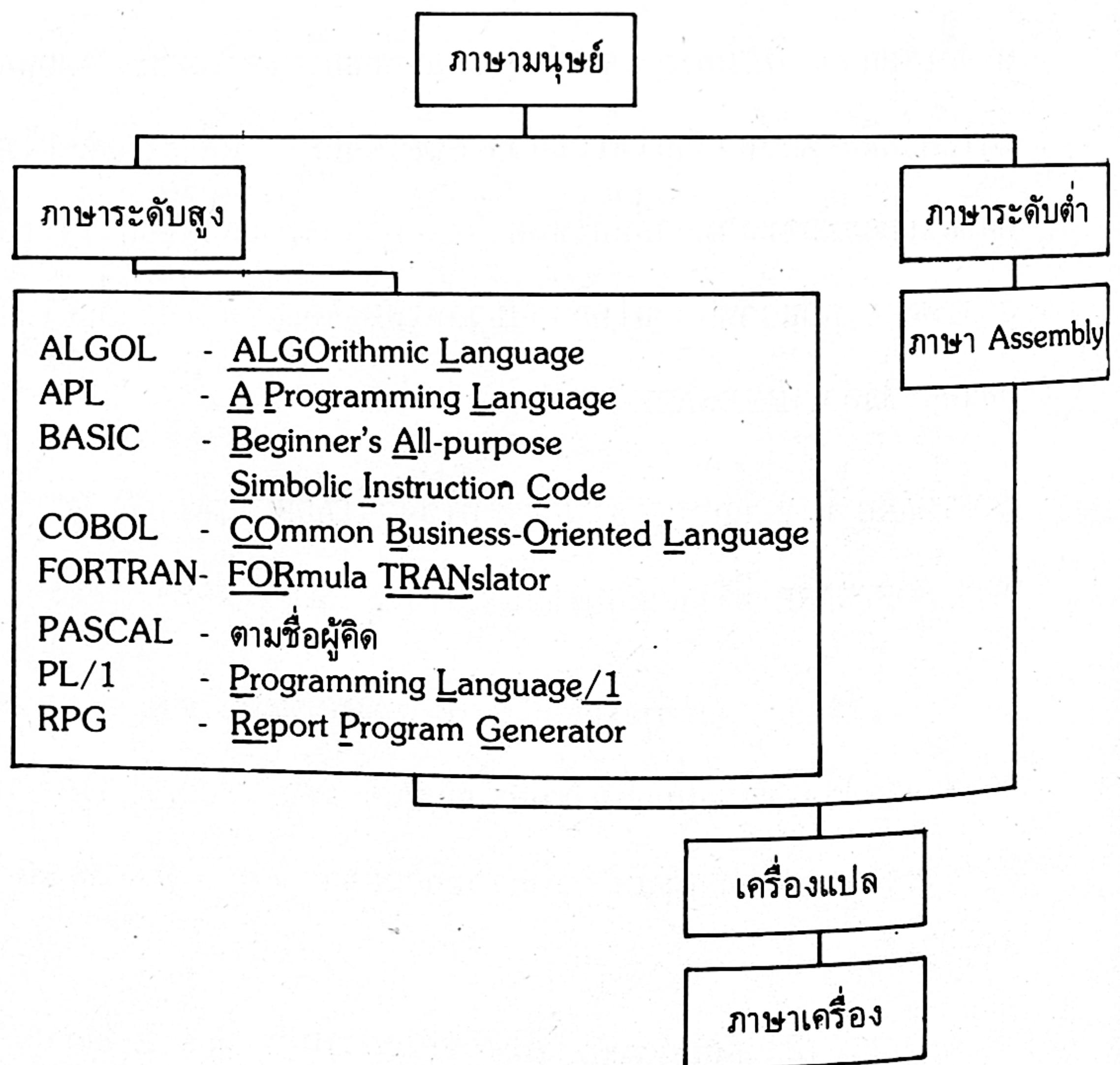
1.5 **SOFTWARE** ดังที่กล่าวมาแล้วว่าอุปกรณ์คอมพิวเตอร์ที่มองเห็นได้ จับต้องได้นั้นเรียกว่า HARDWARE อุปกรณ์เหล่านี้จะเปรียบเสมือนที่ทับกระดาษราคาแพง ทำอะไรไม่ได้เลยหากปราศจากคำชี้แนะให้คอมพิวเตอร์กระทำ หรือที่เรียกว่า *โปรแกรม* ส่วนประกอบที่นอกเหนือไปจาก HARDWARE ที่ทำให้คอมพิวเตอร์ทำงานได้เรียกว่า SOFTWARE ผู้ที่จะเลือกซื้อไมโคร-คอมพิวเตอร์มักจะถามว่ายี่ห้อไหนดี ความจริงแล้วผู้ซื้อควรจะเลือกซื้อ SOFTWARE ก่อนว่าทำงานได้กว้างขวางเพียงไหน หาได้ง่ายแค่ไหน ฯลฯ ก่อนที่จะซื้อตัว HARDWARE

SOFTWARE มี 2 แบบ SOFTWARE ที่ใช้งาน(APPLICATION SOFTWARE) และ SOFTWARE ที่ใช้ควบคุมระบบ(SYSTEM SOFTWARE)

1.5.1 **SOFTWARE ที่ใช้งาน** SOFTWARE แบบนี้มีไว้สำหรับใช้งานตรงตามชื่อ ซึ่งมีอยู่มากมายหลายพันชนิด เช่น เล่นหมากรุก คำนวณขนาดของคาน คิดดอกเบี้ย เก็บรายชื่อสินค้า ฯลฯ SOFTWARE พวกนี้หาซื้อได้ทั่วๆ ไป

1.5.2 **SOFTWARE ที่ใช้ควบคุมระบบ** มีอยู่ 2 ชนิดคือ ระบบการทำงานและเครื่องแปล

- ระบบการทำงาน เช่น แสดงข่าวสารให้ผู้ใช้ทราบว่ากำลังเกิดอะไรขึ้น เช่น เมื่อเปิดเครื่องจะมีอักษรยี่ห้อของเครื่องปรากฏที่ส่วนบนของจอ และมีเครื่องหมายปรากฏขึ้นแล้วบังคับให้เครื่องอ่านบันทึกแผ่นจานหมุน-เพื่อรับข่าวสารมาจากหน่วยความจำภายนอก คือ แผ่นจาน เป็นต้น
- เครื่องแปล ดังที่กล่าวมาแล้วว่าหน่วยความจำภายในของคอมพิวเตอร์จะรับรู้แต่แรงกระตุ้นของประจุไฟฟ้าเท่านั้นว่าเป็นบวกหรือลบ แต่โปรแกรมเขียนเป็นภาษาที่ทำให้คนเข้าใจได้ง่าย เครื่องแปลจะเปลี่ยนโปรแกรมนี้เป็นภาษาเครื่องซึ่งจะสั่งให้เครื่องทำงานได้

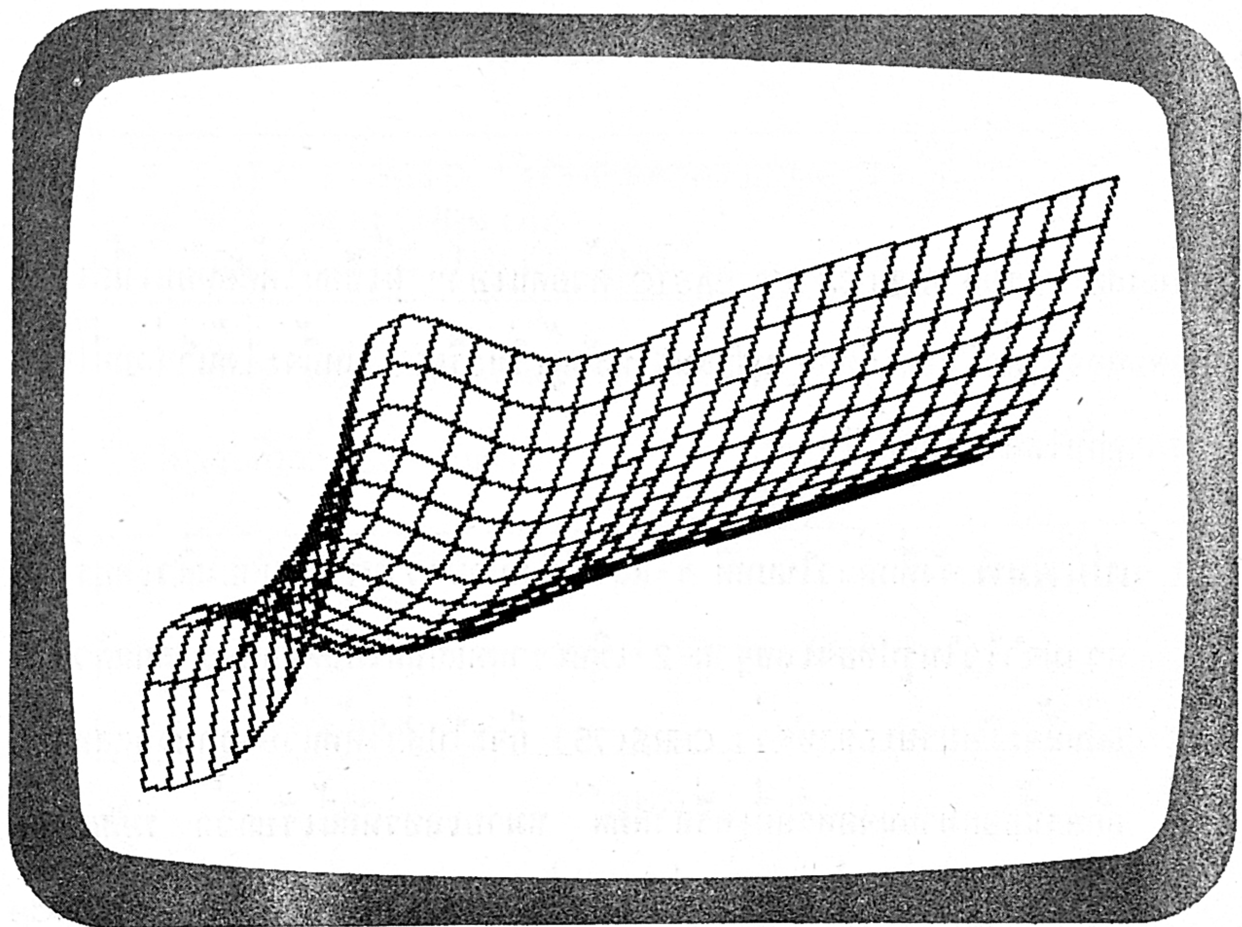


รูปที่ 1.6 การติดต่อทางภาษาระหว่างคนกับเครื่อง

1.6 **ภาษา** การที่คนเราสามารถจะติดต่อกับเครื่องโดยเขียนคำสั่งให้รับรู้เป็นสัญญาณไฟฟ้านั้นเป็นเรื่องยาก จึงมีผู้คิดภาษาขึ้นหลายภาษา แล้วจัดการให้โปรแกรมหรือคำสั่งผ่านเข้าเครื่องแปลแปลออกมาเป็นภาษาเครื่อง เครื่องจะรับรู้และปฏิบัติตามคำสั่งหรือโปรแกรมนั้น รูปที่ 1.6 เป็นแผนภูมิของความสัมพันธ์ระหว่างภาษามนุษย์กับภาษาเครื่อง

1.7 **ภาษา BASIC** ภาษาดังกล่าวมาแล้วต่างก็มีที่ใช้ต่างๆ กัน บางภาษาใช้งานได้ดีในงานธุรกิจ บางภาษาใช้ในงานวิจัย ฯลฯ ภาษา BASIC เป็นภาษาที่ไมโครคอมพิวเตอร์แทบทุกแบบใช้ และใช้ในงานทั่วๆ ไป อาจจะมีข้อจำกัดในการทำงานบางอย่าง แต่อย่างไรก็ดีภาษา BASIC เป็นภาษาที่นิยมใช้กันมากเพราะ

- เป็นภาษาที่ "เป็นกันเอง" เข้าใจง่าย
- มีความคล่องตัว แก้ไขเปลี่ยนแปลงง่าย
- ใช้ได้ทั้งไมโครคอมพิวเตอร์และคอมพิวเตอร์ใหญ่



2 เพิ่มเติมเสริมแต่ง

ในหนังสือ "เรียน APPLESOFT BASIC ด้วยตนเอง" ผู้เขียนได้ตัดทอนเนื้อเรื่องบางตอนออกเพื่อไม่ให้เกิดความสับสนสำหรับผู้เริ่มเรียน บทนี้จะได้นำส่วนที่ได้ยกเว้นมาดูกันให้ละเอียดต่อไป

2.1 **แป้นพิมพ์** ดังที่กล่าวในบทที่ 1 แล้วว่า หน่วยความจำของคอมพิวเตอร์เก็บความจำไว้ในรูปของเลขฐาน 2 เมื่อเรากดแป้นที่เป็นตัวอักษร สมมติว่า K แป้นนี้จะมีหมายเลขรหัสว่า CHR$(75) ก็จะไปสั่งให้หน่วยความจำแสดงตัวอักษรนี้ออกมาบนจอหรือที่เครื่องพิมพ์ หมายเลขรหัสนี้เรียกว่า *รหัสแอสกี้* (ASCII Code) ซึ่งเป็นรหัสที่กำหนดขึ้นโดย American Standard Code for Information Interchange ซึ่งใช้กันเป็นสากล ตารางรหัสนี้แสดงอยู่ในผนวก 1

ดูจากตารางนี้ บางครั้งเราอาจจะใช้การกดแป้นเป็นรหัส แทนที่จะกดบนตัวอักษรที่ต้องการก็ได้ ตัวอย่างเช่น เราอาจจะกด CHR$(38) แทนกดแป้น & ก็ได้ ทดลองกดตัว & จะปรากฏขึ้นมาบนจอภาพ

```
]PRINT CHR$(38)
&
```

ประโยชน์เรื่องนี้ก็มีว่าบางครั้งเราไม่อาจจะกดแป้นพิมพ์ตัวอักษรหรือเครื่องหมายตรงๆ ได้ เช่น ต้องการจะพิมพ์ว่า I SAID "HE IS COMING"

```
]PRINT " I SAID "HE IS COMING"."
 I SAID O.
```

เครื่องจะบอกว่ามีที่ผิด เพราะมีเครื่องหมายคำพูดซ้อนกัน แต่ถ้าพิมพ์ดังนี้

```
]PRINT" I SAID ";CHR$(34);"HE IS COMING"
 ;CHR$(34);CHR$(46)
  I SAID "HE IS COMING".
```

รูปที่ 2.1 จะเป็นโปรแกรมที่จะนำเครื่องหมายบางตัว ซึ่งปกติแล้วเราไม่สามารถพิมพ์ด้วยแป้นพิมพ์ได้

```
10  HOME : REM  PRO#2.1
20  FOR I = 1 TO 480
30  PRINT  CHR$ (91); CHR$ (93);
40  NEXT I
50  GOTO 50
```

รูปที่ 2.1

สังเกตบรรทัดที่ 50 เป็นวงจรที่ถ้าจะตั้งชื่อเป็นไทยก็คงจะได้ว่า *วงจรอนันตร* คือหมุนไปไม่เป็นที่สิ้นสุดจนกว่าเครื่องจะพังไป หรือปิดสวิตช์ หรือกด CTRL C วงจรนี้มีไว้สำหรับไม่ให้เครื่องหมาย PROMPT กับ CURSOR มาโชว์อยู่บนจอภาพให้รกหูรกตาเท่านั้นเอง

2.2 VAL และ STR$

A = 15

คำสั่งนี้เป็นการให้นำเลข 15 เข้าไปบรรจุไว้ในตัวแปรที่มีชื่อ A สภาพของ

ตัวแปรนี้มีค่าเป็นตัวเลขซึ่งจะนำไปบวก ลบ คูณ หาร ฯลฯ อย่างไรก็ได้ รูปที่ 2.2 เป็นการบวกเลขในตัวแปร 2 ตัว

```
10 A = 15
20 B = 12
30  PRINT A + B

]RUN
27
```

รูปที่ 2.2

หากว่าคำสั่งข้างต้นนี้เป็น A$ = "15" สิ่งที่อยู่ในตัวแปรพันธะที่ชื่อ A$ นั้นจะหมดสภาพการเป็นตัวเลขไปทันที ในรูปที่ 2.3 การนำมาบวกกันก็คือนำมาต่อกันนั่นเอง

```
10 A$ = "15"
20 B$ = "12"
30  PRINT A$ + B$

]RUN
1512
```

รูปที่ 2.3

เรามีทางที่จะนำตัวเลขที่อยู่ในตัวแปรพันธะ (ที่หมดสภาพการเป็นตัวเลขไปแล้ว) กลับมาให้มีสภาพเป็นตัวเลขอีก โดยใช้คำสั่ง VAL โปรดดูโปรแกรมในรูปที่ 2.4

```
10 A$ = "15"
20 B$ = "12"
30  PRINT  VAL (A$) +  VAL (B$)
]RUN
27
```

รูปที่ 2.4

ในทางกลับกัน หากเราจะนำตัวเลขมาทำให้หมดสภาพการเป็นตัวเลขไปก็ทำได้ โดยใช้คำสั่ง STR$ ดูโปรแกรมในรูปที่ 2.5

```
10 A = 125
20 B = 250
30  PRINT A + B
40  PRINT  STR$ (A) +  STR$ (B)

]RUN
375
125250
```

รูปที่ 2.5

ลองแกล้งเครื่องดูบ้าง โดยการใช้คำสั่ง VAL กับตัวแปรพันธะที่เป็นตัวอักษร

```
A$ = "AB"
PRINT VAL (A$)
0
```

รูปที่ 2.6

ผลจะออกมาเป็นศูนย์ เพราะตัว AB ไม่มีค่า

2.3 **IF...THEN อีกครั้ง** IF... THEN เป็นคำสั่งที่มีข้อแม้ตามเครื่องหมายเปรียบเทียบ(ดูบทที่ 12 ในเรียน APPLESOFT BASIC ด้วยตนเอง) บางครั้งคำสั่งนี้อาจจะไม่เพียงพอ ดูคำสั่งข้างล่างนี้

IF... AND... THEN

ดูโปรแกรมในรูปที่ 2.7 ถ้าใช้คำสั่ง IF... THEN

```
10  HOME : REM  PRO#2.2
20  INPUT "ENTER YOUR AGE PLEASE
     ";AGE
30  IF AGE < 13 GOTO 60
40  IF AGE > 19 GOTO 60
50  PRINT "HI, TEENAGER": PRINT :
      GOTO 20
60  PRINT "YOU ARE NOT A TEENAGER
     , GOOD BYE": PRINT : GOTO 20

]RUN
ENTER YOUR AGE PLEASE 8
YOU ARE NOT A TEENAGER, GOOD BYE

ENTER YOUR AGE PLEASE 25
YOU ARE NOT A TEENAGER, GOOD BYE

ENTER YOUR AGE PLEASE 15
HI, TEENAGER
```

รูปที่ 2.7

ลองเปลี่ยนบรรทัดที่ 30 ถึง 60 เป็น

```
30  IF AGE > 12 AND AGE < 20 THEN
     50
40  PRINT "YOU ARE NOT A TEENAGER
     , GOOD BYE": PRINT : GOTO 2
    0
50  PRINT : PRINT "HI, TEENAGER "
     : PRINT
60  GOTO 20
```

จะเห็นว่าทั้ง 2 โปรแกรมนี้ให้ผลเหมือนกัน ดูรูปที่ 2.8

```
10  HOME : REM  PRO#2.2MOD.
20  INPUT "ENTER YOUR AGE PLEASE
     ";AGE
30  IF AGE > 12 AND AGE < 20 THEN
     50
```

```
40   PRINT "YOU ARE NOT A TEENAGER
       , GOOD BYE": PRINT : GOTO 20

50   PRINT : PRINT "HI,TEENAGER ":
        PRINT
60   GOTO 20

]RUN
ENTER YOUR AGE PLEASE 8
YOU ARE NOT A TEENAGER, GOOD BYE

ENTER YOUR AGE PLEASE 60
YOU ARE NOT A TEENAGER, GOOD BYE

ENTER YOUR AGE PLEASE 17

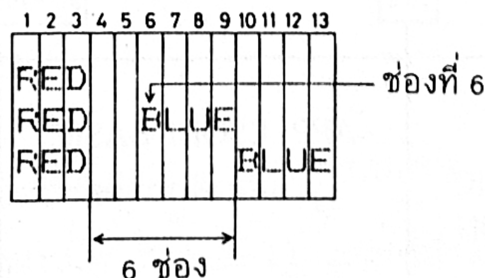
HI,TEENAGER
```

รูปที่ 2.8

2.4 **SPC ( ) กับ TAB ( )** คำสั่ง TAB( ) ที่ตามหลังคำสั่ง PRINT นั้น จะกำหนดตำแหน่งของตัวอักษรตัวแรกที่จะพิมพ์ เช่น

```
10   PRINT "RED"
20   PRINT "RED"; TAB( 6);"BLUE"
30   PRINT "RED"; SPC( 6);"BLUE"

]RUN
RED
RED  BLUE
RED      BLUE
```



รูปที่ 2.9

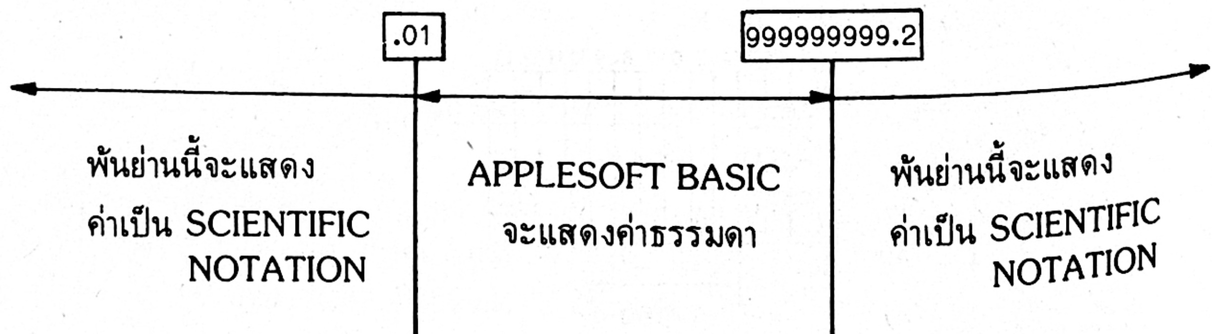
จากตารางในรูปที่ 2.9 ในบรรทัดที่ 20 TAB(6) หมายถึง เริ่มพิมพ์ที่ช่องที่ 6 แต่ SPC(6) หมายถึงว่าให้เว้นช่องไว้ 6 ช่องระหว่างตัวอักษรท้ายของคำแรกจนถึงตัวแรกของคำหลัง

2.5 **SCIENTIFIC NOTATION** ในภาษา APPLESOFT BASIC นั้น เมื่อมีการแสดงค่าของตัวเลขที่มีค่ามากๆ หรือน้อยๆ จะแสดงออกมาในรูปของ SCIENTIFIC NOTATION

เลขธรรมดา	SCIENTIFIC NOTATION	SCIENTIFIC NOTATION ในภาษา APPLESOFT
5,000,000,000	5×10^{9}	5E + 09
.000,005	5×10^{-6}	5E - 06
5 (ตามด้วยเลขศูนย์ 16 ตัว)	5×10^{16}	5E + 16

รูปที่ 2.10

รูปที่ 2.10 แสดงการเปรียบเทียบระหว่างเลขธรรมดากับ SCIENTIFIC NOTATION ที่ใช้กันอยู่และ SCIENTIFIC NOTATION ที่มีอยู่ในภาษา APPLESOFT BASIC



รูปที่ 2.11

ค่าที่ APPLESOFT BASIC จะแสดงเป็นตัวเลขธรรมดาคือ ตัวเลขที่เท่ากับหรือมากกว่า 0.01 จนถึงตัวเลขที่น้อยกว่า 999999999.2 (ตัวเลข 9 มี 9 ตัว) ดูรูปที่ 2.11

เพื่อความเข้าใจ โปรดสังเกตโปรแกรมในรูปที่ 2.12 เปรียบเทียบตัวเลขที่เขียนไว้ในโปรแกรมกับตัวเลขที่ได้เมื่อ RUN ออกมาแล้วและหากพิจารณาร่วมกับรูปที่ 2.11 จะเข้าใจขอบเขตที่ APPLESOFT BASIC ได้กำหนดไว้ให้แสดงเป็นตัวเลขธรรมดา

```
10  REM    PRO#2.4
20  PRINT  123456789
30  PRINT  .0099
40  PRINT  .01
50  PRINT  999999999.19
60  PRINT  999999999.21
70  PRINT  65.02E5

]RUN
123456789
9.9E-03
.01
999999999
1E+09
6502000
```

รูปที่ 2.12

2.6 **แนะนำให้รู้จัก PRINT CHR$ (4)** SYSTEM COMMAND ที่น่าใช้และมีประโยชน์มากคือ CHR$(4)

สอดแผ่นจาน(DISKETTE) ที่ผ่านการ INTIALIZE (ดู "เรียน APPLESOFT BASIC ด้วยตนเอง" หน้า 161) และมีโปรแกรมอยู่บ้างแล้วเข้าไปใน DISK DRIVE แล้วเปิดสวิตช์เครื่อง โปรแกรม HELLO (หรืออย่างอื่น

ตามที่ INITIALIZE ไว้) จะปรากฏบนจอ การกระทำดังนี้เรียกว่า BOOT *DISKETTE*

พิมพ์คำว่า CATALOG ลงไป(อย่าลืมกด RETURN) รายชื่อโปรแกรมที่บันทึกไว้จะปรากฏขึ้นมา

พิมพ์ HOME รายการนั้นจะหายไป

พิมพ์ NEW แล้วพิมพ์โปรแกรมนี้ลงไป

```
10 PRINT CHR$(4);"CATALOG"
```

แล้ว RUN

รายชื่อโปรแกรม(CATALOG) จะปรากฏขึ้นมา

คำสั่ง PRINT CHR$(4) คือ คำสั่งที่เมื่อบรรจุไว้ในโปรแกรมแล้ว เราสามารถสั่งให้เครื่องดำเนินการเรียก CATALOG มาให้ดู แล้ว LOAD หรือ RUN โปรแกรมอื่นต่อเนื่องกันไปทันที

ลองเขียนโปรแกรมง่ายๆ นี้ลงไป

```
10 PRINT "ARE YOU SURE YOU UNDERSTAND THIS
 SUBJECT"
```

แล้ว SAVE โปรแกรมนี้ โดยการพิมพ์ SAVE AREYOUSURE คำตามหลังคำว่า SAVE คือ ชื่อของ FILE (FILE NAME)

พิมพ์อีกโปรแกรมลงไป

```
NEW
10 PRINT "I AM GOING TO EXPLAIN YOU
   ABOUT USING THIS COMMAND"
20 PRINT
30 PRINT CHR$(4);"RUN AREYOUSURE"
```

SAVE โปรแกรมนี้ไว้ให้ชื่อว่า EXPLAIN

พิมพ์ CATALOG แล้วตรวจชื่อรายชื่อโปรแกรมที่มีอยู่ จะเห็นว่ามีโปรแกรมเพิ่มขึ้น 2 โปรแกรมคือ AREYOUSURE กับ EXPLAIN

พิมพ์ RUN EXPLAIN จะมีภาพปรากฏบนจอดังนี้

```
I AM GOING TO EXPLAIN YOU ABOUT USING THIS
COMMAND
ARE YOU SURE YOU UNDERSTAND THIS SUBJECT
```

เครื่องจะ RUN แล้วพิมพ์คำสั่งตามบรรทัดที่ 10 ออกมา พอถึงบรรทัดที่ 30 จะมีคำสั่งให้ RUN อีกโปรแกรมหนึ่ง (คือ AREYOUSURE) ทันที

เราอาจจะใช้ [CTRL] D แทน CHR$(4) ได้ โดยการพิมพ์ดังนี้

```
30 PRINT"";"RUN AREYOUSURE"
```

หลังจากพิมพ์คำว่า PRINT "แล้วให้กด [CTRL] D แล้วจึงพิมพ์ต่อไปคือ " ; "RUN AREYOUSURE"

คำสั่งนี้จะมีประโยชน์มาก เพราะสามารถจะทำให้เราเรียกใช้โปรแกรมอื่นที่อยู่ในแผ่นจาน(DISKETTE) ต่อไปได้ทันที โดยไม่ต้อง LOAD และ RUN ใหม่

2.7 **ONERR GOTO...., ONERR GOSUB...** ขณะที่เครื่องดำเนินไปตามโปรแกรม บางครั้งอาจจะมีข้อผิดพลาด(ERROR) ในโปรแกรม จะโดยตั้งใจหรือไม่ตั้งใจก็ตามแต่เครื่องจะหยุดทำงานทันที และบอกออกมาบนจอภาพว่าเกิดขึ้นเพราะเหตุใด

กรณีที่เราต้องการให้เครื่องดำเนินต่อไป ถึงแม้จะมี ERROR เกิดขึ้น เราก็สามารถทำได้โดยใช้คำสั่งข้างต้น

ดูตัวอย่างในโปรแกรมรูปที่ 2.13 หากมีการนำเลขศูนย์มาหารแล้วจะต้องเกิดข้อผิดพลาด(ERROR) แต่เรามีโอกาสให้แก้ตัวใหม่ได้

```
10  REM  PRO#2.5
20  INPUT "A= ";A
30  INPUT "B= ";B
40  ONERR  GOTO 70
50 C = A / B
60  PRINT : PRINT "A/B= ";C: END
70  PRINT "VALUE OF'B'SHOULD NOT
     BE ZERO,TRY AGAIN
80  GOTO 30


]RUN
A= 6
B= 0
VALUE OF'B'SHOULD NOT BE ZERO,TRY AGAIN
B= 2
A/B= 3
```

รูปที่ 2.13

ลอง RUN PROGRAM ดู แล้วให้ค่าของ B ที่ไม่เท่ากับศูนย์ โปรแกรมจะดำเนินมาจนจบ ลอง RUN อีกครั้งแล้วให้ค่าของ B เท่ากับศูนย์ แล้วสังเกตผลที่ออกมา บรรทัดที่ 40 จะเป็นตัวตรวจสอบหากมีความผิดพลาด(ERROR) ก็จะกระโดดไปบรรทัดที่ 70

ข้อพึงสังเกต คำสั่งนี้จะต้องอยู่ก่อนบรรทัดที่จะมีการคำนวณที่จะทำให้เกิด ERROR ขึ้น

2.8 **YES หรือ NO** เรื่องที่น่าสนุกประการหนึ่งในการ *เล่น* กับคอมพิวเตอร์ คือเราสามารถเขียนโปรแกรมให้โต้ตอบกับเราได้ ที่ใช้กันมากคือคอมพิวเตอร์จะถามเราว่า YES หรือ NO คำตอบที่เราจะให้จะต้องตรงกับที่เขียนโปรแกรมไว้ มิฉะนั้นคอมพิวเตอร์จะไม่ยอมทำงาน ตัวอย่างตามรูปที่ 2.14 ข้างล่างเป็นโปรแกรมง่ายๆ ที่สามารถให้คอมพิวเตอร์บอกเราว่าเราตอบถูกหรือผิด

```
10  REM  PRO#2.6
15  HOME
20  INPUT "DO YOU LIKE THIS SHIRT
      ?";A$
30  IF  LEFT$ (A$,1) = "Y" THEN 6
     0
40  IF  LEFT$ (A$,1) = "N" THEN 7
     0
45  PRINT
50  PRINT "PLEASE ANSWER YES OR N
     O.": GOTO 20
60  PRINT : PRINT "GOOD,IT'S YOUR
     S": GOTO 80
70  PRINT : PRINT " TOO BAD,I INT
     ENDED TO GIVE IT TO YOU": GOTO
     80
80  END

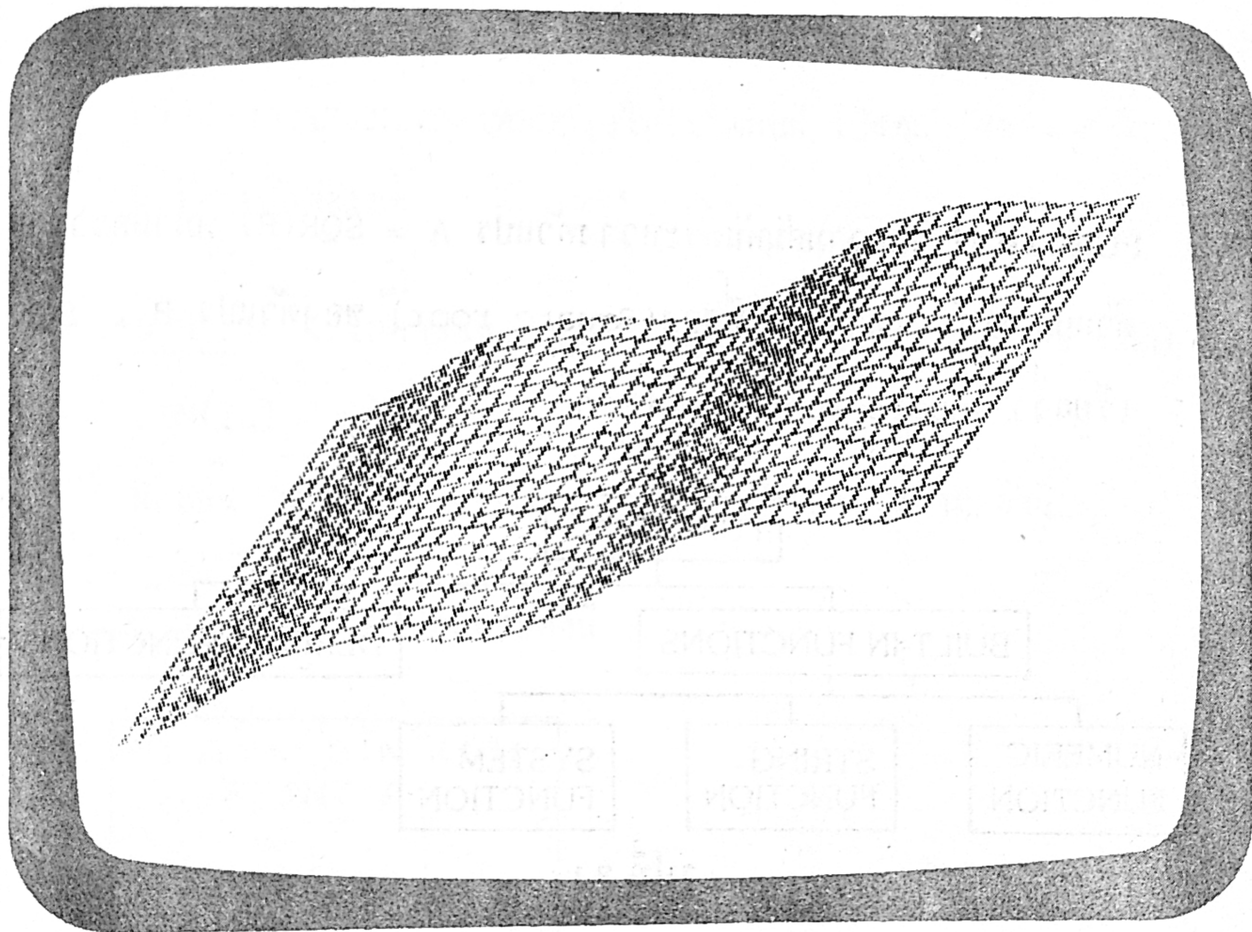

]RUN
DO YOU LIKE THIS SHIRT ?O.K.

PLEASE ANSWER YES OR NO.
DO YOU LIKE THIS SHIRT ?Y

GOOD,IT'S YOURS
```

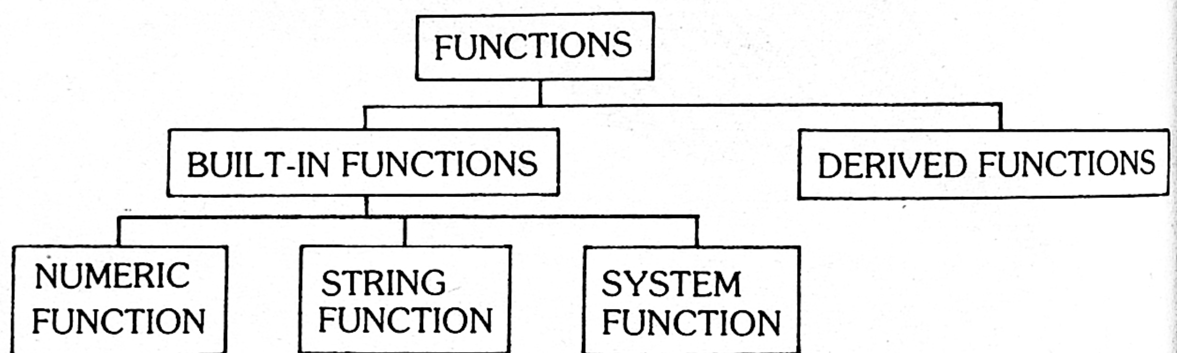
รูปที่ 2.14

ไม่ว่าเราจะตอบ YES หรือ Y, NO หรือ N เครื่องจะกระโดดไปบรรทัดที่ 60 หรือ 70 แต่ถ้าตอบไม่ตรงอย่างหนึ่งอย่างใดตามที่กล่าวมาเลย เครื่องก็จะคาดคั้นให้เราให้คำตอบจนได้ (ดูคำสั่ง LEFT$ ใน "เรียน APPLE-SOFT BASIC ด้วยตนเอง" หน้า 85) จำตัวอย่างนี้ไปดัดแปลงใช้กับคำถามอื่นได้



3 FUNCTIONS

3.1 **FUNCTION** คือความสัมพันธ์ระหว่างตัวแปร A = SQR(B) หมายความว่า ตัวแปร A มีค่าเท่ากับรากที่สอง(Square root) ของตัวแปร B , SQR เรียกว่า Square root function



รูปที่ 3.1

FUNCTION ที่ใช้ในไมโครคอมพิวเตอร์มีอยู่ 2 แบบ(ดูรูปที่ 3.1) แบบแรกเรียกว่า BUILT-IN FUNCTION หรือ LIBRARY FUNCTION หรือ STANDARD FUNCTION หรือ ELEMENTARY FUNCTION คือ FUNCTION ที่ผู้สร้างเครื่องได้บรรจุไว้ในเครื่อง ผู้ใช้สามารถจะเรียกมาใช้ได้ทันที ส่วนอีกแบบหนึ่งเรียกว่า DERIVED FUNCTION หรือ USER-DEFINED FUNCTION คือ FUNCTION ที่ผู้ใช้สามารถจะสร้างขึ้นเอง บรรจุไว้ในโปรแกรมแล้วสามารถเรียกมาใช้ได้

3.2 **BUILT-IN FUNCTIONS** สามารถที่จะใช้ได้ทันทีหรือบรรจุไว้ในโปรแกรมก็ได้มี 3 แบบคือ

3.2.1 **SYSTEM FUNCTION** แบบนี้สำหรับผู้ที่เรียนรู้โปรแกรมจนเชี่ยวชาญแล้ว จึงไม่ขอกล่าวในที่นี้

3.2.2 **NUMERIC FUNCTION** ใน APPLESOFT BASIC ได้มีการสร้าง FUNCTION เหล่านี้ไว้ให้ใช้ได้ทันที โปรดสังเกตว่า X ในวงเล็บจะมีค่าเป็นตัวเลข

SIN(X) จะให้ค่าของ sine ของมุม X (ซึ่งมีหน่วยเป็น Radian) เช่น SIN(14)

กดแป้น]PRINT SIN(14) RETURN จะให้ค่ามาดังนี้ .990607356

ถ้าจะบรรจุไว้ในโปรแกรมดังนี้

```
10 A =  SIN (14)
20  PRINT A

]RUN
.990607356
```

COS(X) จะให้ค่าของ Cosine ของมุม X (Radian)

```
]PRINT COS(16)
-.957659481
```

TAN(X) จะให้ค่าของ Tangent ของมุม X (Radian)

```
]PRINT TAN(.4)
.422793219
```

ATN(X) จะให้ค่าของ Arctangent เป็นมุม Radian ซึ่งมุมนี้จะมีค่าระหว่าง $-\frac{\pi}{2}$ ถึง $+\frac{\pi}{2}$

```
]PRINT ATN(4)
1.32581766
```

INT(X) จะให้ค่าเลขจำนวนเต็มของ X (คือ ไม่มีเลขหลังทศนิยม) หรืออีกนัยหนึ่งคือ การตัดเศษทิ้งไป

```
]PRINT INT(3.2)
3

]PRINT INT(3.9)
3
```

RND(X) จะให้ค่าตัวเลขสุ่ม(RANDOM NUMBER) ซึ่งตัวเลขสุ่มนี้จะมีค่าอยู่ระหว่าง 0 ถึง 1

- ถ้า X มากกว่า 0 RND(X) จะให้ตัวเลขใหม่ทุกครั้งไม่ซ้ำกัน

```
]PRINT RND(3)
.973136996

]PRINT RND(3)
.103117626
```

- ถ้า X น้อยกว่า 0 ในค่า X จำนวน เดียวกัน RND(X) จะให้ค่าเลขจำนวนเดียวกัน

```
]PRINT RND(-4)
2.99214662E-08

]PRINT RND(-4)
2.99214662E-08
```

- ถ้า X เท่ากับ 0 RND(X) จะให้ค่าค่าเดียวกับค่าสุดท้ายที่ได้ทำมาแล้ว

```
]PRINT RND(4)
.808563215

]PRINT RND(4)
.236586843

]PRINT RND(0)
.236586843
```

(เหมือนกัน)

<u>SGN(X)</u> จะให้เครื่องหมายของ X

- ถ้า X > 0 SGN(X) = 1
- ถ้า X < 0 SGN(X) = -1
- ถ้า X = 0 SGN(X) = 0

เช่น

```
]PRINT SGN(3.5)
1

]PRINT SGN(-5)
-1

]PRINT SGN(0)
0
```

<u>ABS(X)</u> จะให้ค่าสมบูรณ์(ABSOLUTE) ของ X

```
]PRINT ABS(-4)
4

]PRINT ABS(3.5)
3.5
```

SQR(X) จะให้ค่ารากที่สอง (Square root) ของ X (ที่เป็นบวก)

```
]PRINT SQR(8)
2.82842713

]PRINT SQR(-6)

?ILLEGAL QUANTITY ERROR
```

EXP(X) จะให้ค่า e^x (e = 2.718289)

```
]PRINT EXP(2)
7.3890561
```

LOG(X) จะให้ค่า log ฐาน e ($\log_e$ (X))

```
]PRINT LOG(8)
2.07944154
```

3.2.3 **STRING FUNCTION** ในหัวข้อ 3.2.2 ตัวในวงเล็บจะมีค่าเป็นตัวเลขใน STRING FUNCTION ที่สร้างไว้ในภาษา APPLESOFT BASIC สามารถเรียกใช้ได้ทันทีนั้นตัวในวงเล็บอาจจะเป็นตัวเลข หรือไม่เป็นตัวเลขก็ได้ FUNCTION พวกนี้ใช้ได้โดยตรงทันที หรือจะบรรจุในโปรแกรมก็ได้

CHR$(X) ได้กล่าวไว้แล้วใน 2.1

ASC(" ") จะให้รหัสแอสกี้(ASCII Code) ของตัวอักษรที่อยู่ในเครื่องหมายคำพูด(หัวข้อที่ 2.1 และภาคผนวก)

```
]PRINT ASC("M")
77
```

หมายถึงว่าอักษร M มีรหัสแอสกี้ 77

VAL(X$) ได้กล่าวไว้แล้วในหัวข้อ 2.2

STR$(X) ได้กล่าวไว้แล้วในหัวข้อ 2.2

LEN(X$) จะให้ค่าเป็นจำนวนตัวอักษรที่ประกอบเป็น X$

LEFT$(X$, X) จะแสดงตัวอักษรที่อยู่ซ้ายมือสุดของ X$ เป็นจำนวน X ตัวอักษร

RIGHT$(X$, X) จะแสดงตัวอักษรที่อยู่ขวามือสุดของ X$ เป็นจำนวน X ตัวอักษร(นับจากขวามือมา)

MID$(X$, X, Y) จะแสดงตัวอักษรที่อยู่ตรงกลางของ X$ นับตั้งแต่ X ตัวอักษรจากซ้ายมือไปเป็นจำนวน Y ตัวอักษร ดูรูปที่ 3.2

```
10  REM  PRO#3.1
20 A$ = "APPLESOFT"
30  PRINT : PRINT A$
40  PRINT : PRINT  LEN (A$)
50  PRINT : PRINT  LEFT$ (A$,4)
60  PRINT : PRINT  RIGHT$ (A$,3)
70  PRINT : PRINT  MID$ (A$,4,3)
80  END

]RUN

APPLESOFT

9
```

```
APPL

OPT

LES
```

รูปที่ 3.2

3.3 **DERIVED FUNCTION หรือ USER-DEFINED FUNCTION** เป็น FUNCTION ที่สามารถเรียกมาใช้ได้ แต่ FUNCTION นี้มิได้สร้างมากับเครื่อง หากแต่เป็น FUNCTION ที่ผู้ใช้สร้างขึ้นเอง ซึ่งจะมีประโยชน์มากในกรณีที่เป็นสูตรที่ต้องใช้มากกว่า 1 ครั้ง

สมมติว่ามีสูตรหาพื้นที่ของวงกลมที่ต้องใช้บ่อยๆ เราจะกำหนดสูตรนี้ไว้ในหน่วยความจำ โดยใช้คำสั่งดังนี้

```
DEF  FN A(R)  =  3.14159 * R ∧ 2
```

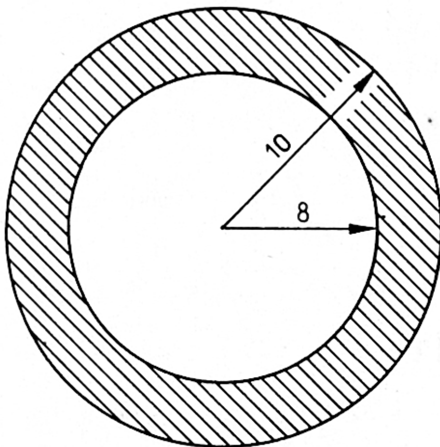
A คือชื่อของ FUNCTION

R เรียกว่า ตัวอ้างอิง(ARGUMENT) ซึ่งตัวที่ตามหลังเครื่องหมายเท่ากับจะขึ้นอยู่กับตัวอ้างอิงนี้

```
10  REM  PRO#3.2
20  DEF  FN A(R) = 3.14159 * R ^
     2
30  INPUT "RADIUS OF OUTER CICLE=
     ";M
40  INPUT "RADIUS OF INNER CIRCLE
     = ";N
50  PRINT "AREA OF RING= "; FN A(
     M) -  FN A(N)
```

```
]RUN
RADIUS OF OUTER CICLE= 15
RADIUS OF INNER CIRCLE= 9
AREA OF RING= 452.38896
```

รูปที่ 3.3



ตามโปรแกรมในรูปที่ 3.3

บรรทัดที่ 20 กำหนดสูตรของพื้นที่ของรูปวง-กลมโดยกำหนดให้มีรัศมี(R) เป็น "ตัวอ้างอิง" (ARGUMENT)

บรรทัดที่ 30 และ 40 เป็นการป้อนข้อมูล ซึ่งเป็นรัศมีของวงกลมนอกและวงกลมใน

บรรทัดที่ 50 ให้พิมพ์ค่าที่ได้จากผลลบของวง-กลมนอกและวงกลมใน

สรุป

FUNCTION มี 2 แบบคือ BUILT-IN FUNCTION เป็น FUNCTION ที่สร้างไว้ในเครื่องแล้ว กับ DERIVED FUNCTION เป็น FUNCTION ที่ผู้ใช้สามารถสร้างเองได้

ข้างล่างนี้คือ BUILT-IN FUNCTION ที่สร้างไว้ใน APPLESOFT BASIC

SIN	COS	TAN	ATN	INT	RND
SGN	ABS	SQR	EXP	LOG	CHR$
ASC	VAL	STR$	LEN	LEFT$	RIGHT$
MID$					

แบบฝึกหัดบทที่ 3

3.1 รูปเสี้ยวของวงกลม มีสูตรการคำนวณดังนี้

$$L = 0.01745\ RN$$

$$A = 0.008727\ NR^2$$

โดยที่ L = ความยาวของส่วนโค้ง

R = รัศมีของวงกลม

N = มุมของรูปเสี้ยวเป็นองศา

A = พื้นที่ของรูปเสี้ยว

จงทำตารางหาส่วนโค้ง และพื้นที่ของรูปเสี้ยวของวงกลมที่มีรัศมี R = 25 และมุมเปลี่ยนแปลงตั้งแต่ 20 ถึง 30 องศา

3.2 จงหาค่าของ SINE และ COSINE ของมุมตั้งแต่ 1 องศาจนถึง 10 องศา โดยการใช้ BUILT-IN FUNCTION และ DERIVED FUNCTION (π RADIAN = 180 องศา)

เฉลยแบบฝึกหัดบทที่ 3

3.1

```
20  HOME : REM    SOL#3.1
30  DEF  FN L(N) = .01745 * 25 *
     N
40  DEF  FN A(N) = .008727 * N *
     25 ^ 2
45  PRINT "ANGLE"; TAB( 10);"ARC
     LENGTH"; TAB( 25);"AREA": PRINT

50  FOR N = 20 TO 30
60  PRINT  TAB( 2);N; TAB( 10); FN
     L(N); TAB( 25); FN A(N)
70  NEXT N

]RUN
ANGLE     ARC LENGTH     AREA

 20       8.725          109.0875
 21       9.16125        114.541875
 22       9.5975         119.99625
 23       10.03375       125.450625
 24       10.47          130.905
 25       10.90625       136.359375
 26       11.3425        141.81375
 27       11.77875       147.268125
 28       12.215         152.7225
 29       12.65125       158.176875
 30       13.0875        163.63125
```

3.2

```
10  HOME : REM    SOL#3.2
15  DIM S(10),C(10)
20  DEF  FN B(A) = 3.14159 / 180 *
     A
30  PRINT "ANGLE"; TAB( 10)"SINE"
     ; TAB( 25)"COSINE": PRINT
40  FOR I = 1 TO 10
60 S(I) =  SIN ( FN B(I))
70 C(I) =  COS ( FN B(I))
80  PRINT I; TAB( 10)S(I); TAB( 2
     5)C(I): PRINT
90  NEXT I


]RUN
ANGLE      SINE               COSINE

1          .0174523917        .999847695

2          .0348994672        .999390828

3          .0523359121        .998629537

4          .0697564149        .997564054

5          .0871556693        .996194704

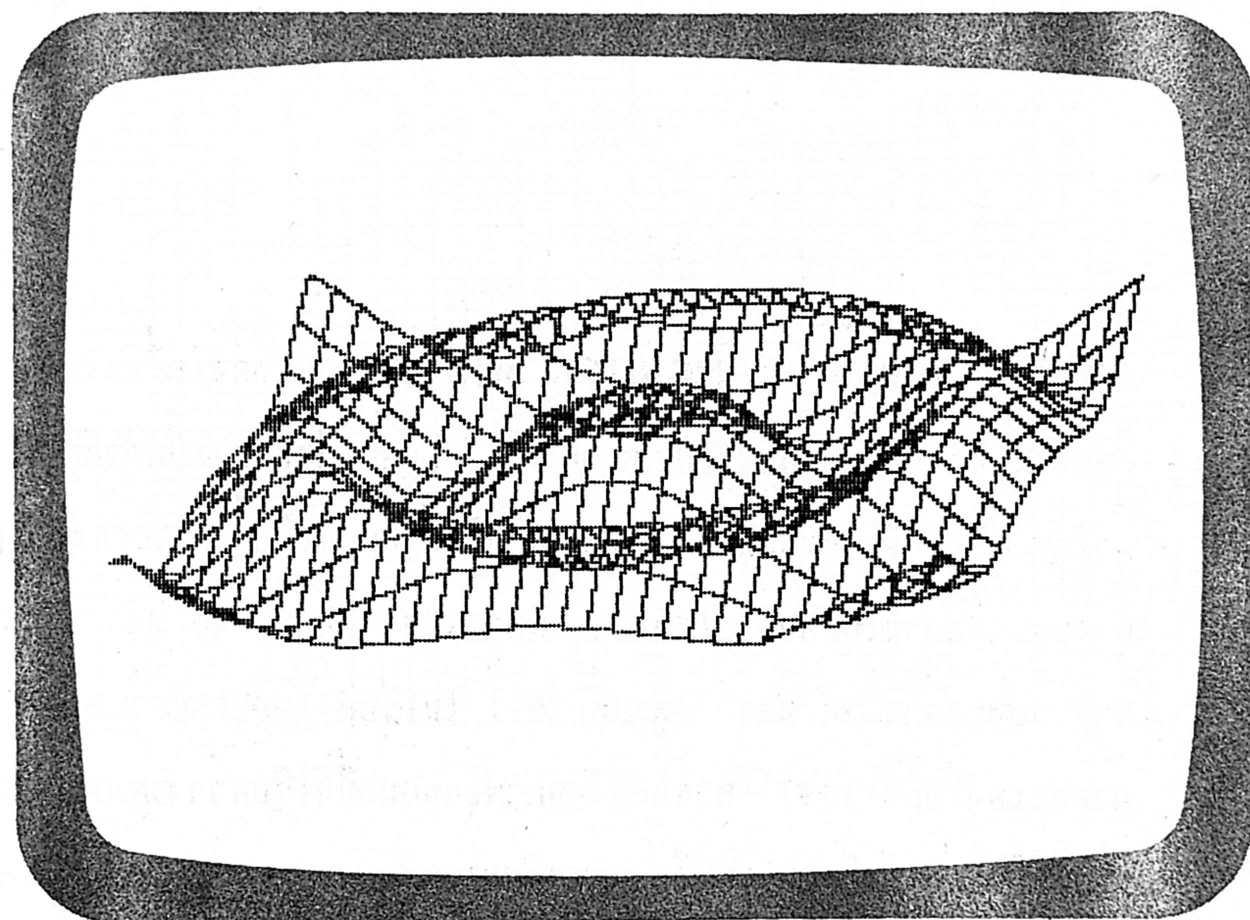
6          .104528375         .994521905

7          .121869241         .992546164

8          .139172984         .990268085

9          .156434334         .987688361

10         .173648033         .984807779
```



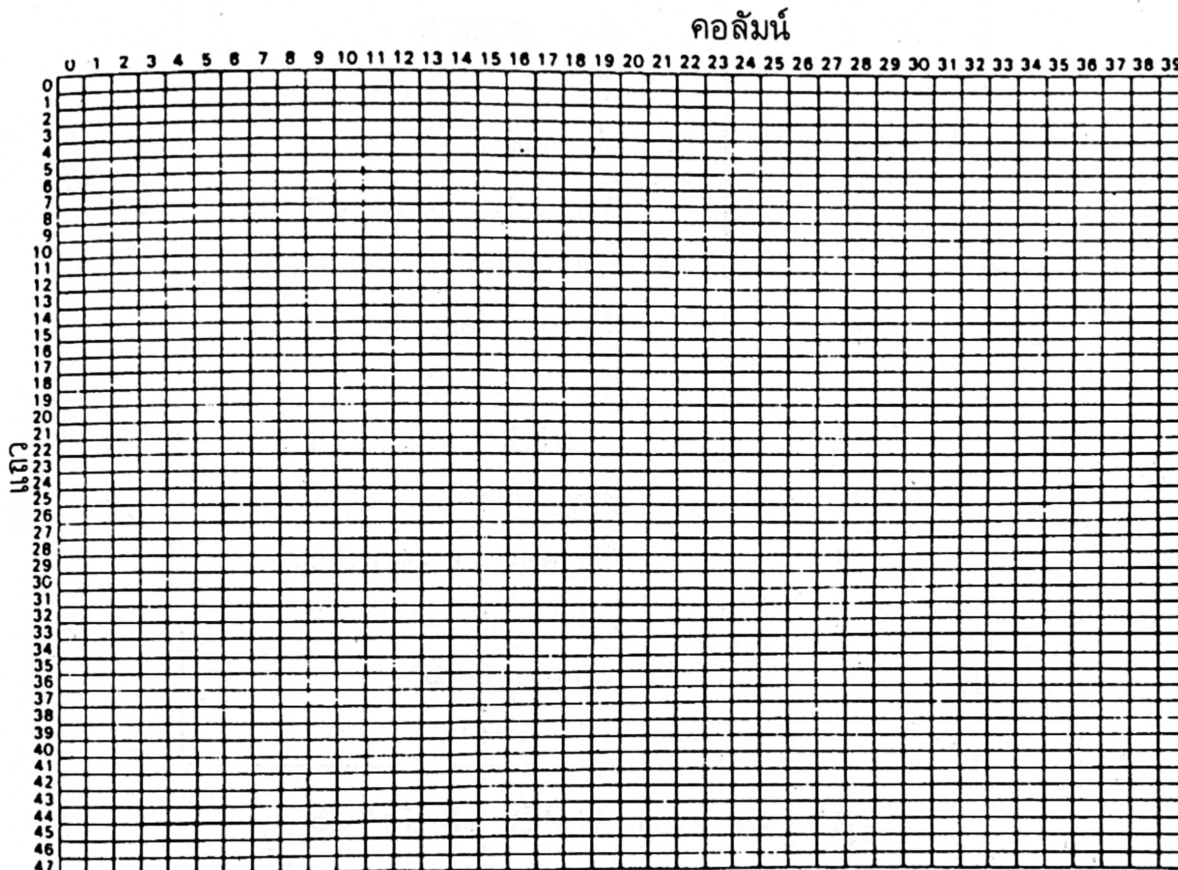
4 กราฟิก

4.1 **ภาพที่แสดงบนจอภาพ (MONITOR หรือ CRT)** ของเครื่องไมโครคอมพิวเตอร์จะแสดงออกมาเป็นสองแบบคือ แบบหนึ่งแสดงเป็นตัวอักษร, ตัวเลข หรือเครื่องหมายที่ปรากฏอยู่บนแป้นพิมพ์ แบบนี้เรียกว่า *TEXT MODE* ถ้าจะแบ่งจอภาพเป็นตารางจะมีช่องจากซ้ายไปขวาทั้งหมด 40 ช่อง และจากบนลงมาล่าง 24 ช่อง (ดูรูปที่ 18.1 ในเรียน APPLESOFT BASIC ด้วยตนเอง หน้า 109) การแสดงจอภาพอีกแบบหนึ่งเรียกว่า *GRAPHICS MODE* ซึ่งแบ่งเป็นอีก 2 แบบคือ *กราฟิกแบบเส้นหยาบ* (LOW RESOLUTION GRAPHICS) และ *กราฟิกแบบเส้นละเอียด* (HIGH RESOLUTION GRAPHICS) ซึ่งจะใช้ประโยชน์ได้มากในการแสดงบนจอภาพเป็นภาพต่างๆ

4.2 **กราฟิกเส้นหยาบ (LOW RESOLUTION GRAPHICS)** หรือที่เรียกย่อๆว่า LO-RES GRAPHICS) แบ่งเป็น 2 แบบคือ SPLIT SCREEN GRAPHICS กับ FULL SCREEN GRAPHICS

ใน FULL SCREEN GRAPHICS จอภาพจะแบ่งออกเป็นตาราง(ซึ่งมองไม่

เห็น) จากซ้ายไปขวาจะแบ่งออกเป็น 40 คอลัมน์ (เหมือน TEXT MODE) แต่จากบนลงล่างจะแบ่งออกเป็น 48 แถว ดูรูปที่ 4.1



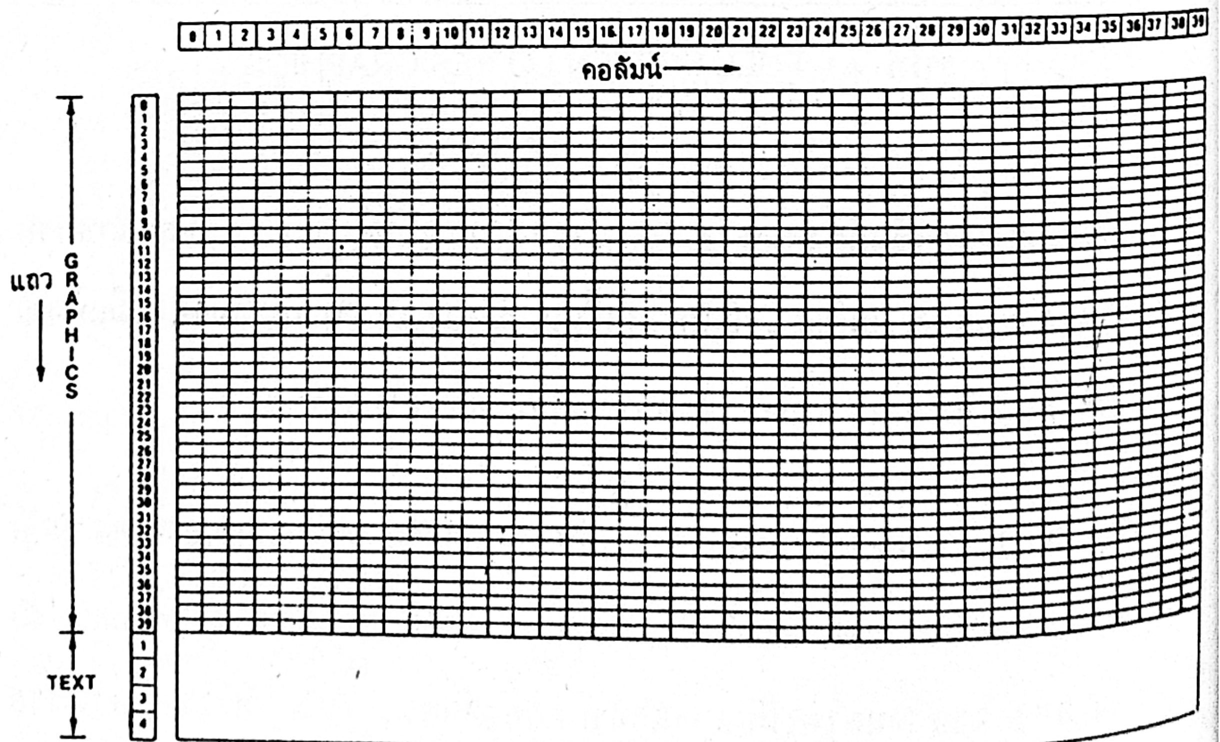
รูปที่ 4.1 FULL SCREEN LO-RES GRAPHICS

ในแต่ละตารางซึ่งมีทั้งหมด 1920 ตาราง(40 × 48) นั้น เราสามารถกำหนดให้แสดงเป็นสีอะไรก็ได้ สีทั้งหมดมี 16 สี (ถ้าจอภาพของเราเป็นจอภาพสี) สีดังกล่าวนี้มีหมายเลขต่างกันดังแสดงในรูปที่ 4.2

ใน SPLIT SCREEN GRAPHICS นั้น การแบ่งช่องของจอภาพคล้ายกับ FULL SCREEN GRAPHICS เพียงแต่ว่าใน 8 แถวสุดท้าย (คือแถว 40 ถึง 47) จะกำหนดให้เป็นช่องสำหรับแสดงอักษร, เลข หรือเครื่องหมาย ซึ่งมีชื่อเรียกว่า *TEXT WINDOW* และมีจำนวนทั้งหมด 4 บรรทัดดูรูปที่ 4.3

หมายเลข	สี
0	ดำ (BLACK)
1	แมกเยนต้า (MAGENTA)
2	น้ำเงินแก่ (DARK BLUE)
3	ม่วง (PURPLE)
4	เขียวแก่ (DARK GREEN)
5	เทาเบอร์ 1 (GRAY #1)
6	น้ำเงินเข้ม (MEDIUM BLUE)
7	น้ำเงินอ่อน (LIGHT BLUE)
8	น้ำตาล (BROWN)
9	ส้ม (ORANGE)
10	เทาเบอร์ 2 (GRAY #2)
11	ชมพู (PINK)
12	เขียวอ่อน (LIGHT GREEN)
13	เหลือง (YELLOW)
14	แอกคว่า (AQUA)
15	ขาว (WHITE)

รูปที่ 4.2 สีของ LO-RES GRAPHICS



รูปที่ 4.3 SPLIT SCREEN LO-RES GRAPHICS

4.2.1 เราจะเข้าไปใน LO-RES GRAPHICS MODE ได้อย่างไร

เมื่อเริ่มเปิดสวิตช์เครื่องไมโครคอมพิวเตอร์บนจอภาพจะเป็น TEXT MODE คือ สามารถจะพิมพ์ตัวอักษร ตัวเลข หรือเครื่องหมายบนแป้นพิมพ์

พิมพ์ GR [RETURN]

ขณะนี้เครื่องจะเข้า LO-RES GRAPHICS MODE ภาพต่างๆ ส่วนบนของจอภาพจะหายไปหมด(หรือมีสีดำทั้งจอ) โปรแกรมหรือคำสั่งที่จะพิมพ์ต่อจากนี้ไปจะไม่ปรากฏที่ส่วนบนของจอภาพ นอกจากจะสั่งให้ไปปรากฏที่ส่วนล่างของจอภาพเพียง 4 บรรทัด ซึ่งต่อไปนี้จะเรียกว่า TEXT WINDOW และตอนนี้จะเป็น SPLIT SCREEN GRAPHICS

หากจะให้เครื่องไปอยู่ใน FULL SCREEN GRAPHICS (ดูหมายเหตุท้ายข้อนี้) ให้พิมพ์

POKE - 16302, 0

เมื่อต้องการกลับไปใน SPLIT SCREEN GRAPHICS ก็เพียงแต่กด GR

ถ้าต้องการกลับไปใน TEXT MODE คืออยู่ในสภาพที่ใช้งานตามปกติ ให้พิมพ์ TEXT บนจอภาพจะปรากฏภาพตัวหนังสือ หรือเครื่องหมายเต็มจอ เมื่อพิมพ์ HOME ภาพบนจอจะหายไป

การเข้าไปใน LO-RES GRAPHICS MODE เราจะเข้าได้ทันที(IMMEDIATE EXECUTION) หรือบรรจุไว้ในโปรแกรมก็ได้

เมื่อกด GR RETURN เครื่องจะเข้า LO-RES GRAPHICS MODE แบบ SPLIT SCREEN ทันที ต่อจากนั้นไปเมื่อพิมพ์อะไรลงบนจอจะปรากฏอยู่บน 4 บรรทัดสุดท้ายเท่านั้น

เมื่อบรรจุ GR เข้าไปในโปรแกรม(เป็น DEFERRED EXECUTION) เราสามารถจะพิมพ์โปรแกรมไปได้ตามปกติธรรมดา เครื่องจะเข้า GRAPHICS MODE ก็ต่อเมื่อ RUN PROGRAM แล้วเท่านั้น และจะคงอยู่ใน GRAPHICS MODE จนกว่าจะมีคำสั่ง TEXT ซึ่งอาจจะสั่งทันทีหรือสั่งในโปรแกรมก็ได้

หมายเหตุ การใช้ FULL SCREEN GRAPHICS MODE นั้น ในระยะเริ่มแรกอาจจะมีข้อปลีกย่อยที่จะทำให้สับสนแก่ผู้เริ่มเรียน ดังนั้นตั้งแต่นี้ไปจะได้กล่าวถึง SPLIT SCREEN GRAPHICS เท่านั้น

4.2.2 ดังที่กล่าวมาแล้วว่าใน LO-RES GRAPHICS MODE นั้น เราสามารถกำหนดสีในแต่ละตารางของ 1920 ตารางได้ทั้งหมด 16 สีคือ จากสี 0 จนถึงสี 15 ทั้งนี้เราจะเห็นได้ชัดหากใช้จอภาพเป็นแบบสี แต่ถ้าจอภาพเป็นขาว-ดำจะเห็นการแตกต่างกันไม่มากนัก เมื่อเราพิมพ์ GR ให้เครื่องเข้าใน LO-RES GRAPHICS MODE ทุกตารางจะมีสี 0 คือสีดำโดยอัตโนมัติ ดังนั้นไม่ว่าจะเขียนอะไรลงไปก็จะปรากฏเป็นสีดำหมด ซึ่งจะมองไม่เห็นบนจอ ดังนั้นจึงจำเป็นที่จะต้องกำหนดสีทันทีหลังจากการกด GR เช่น

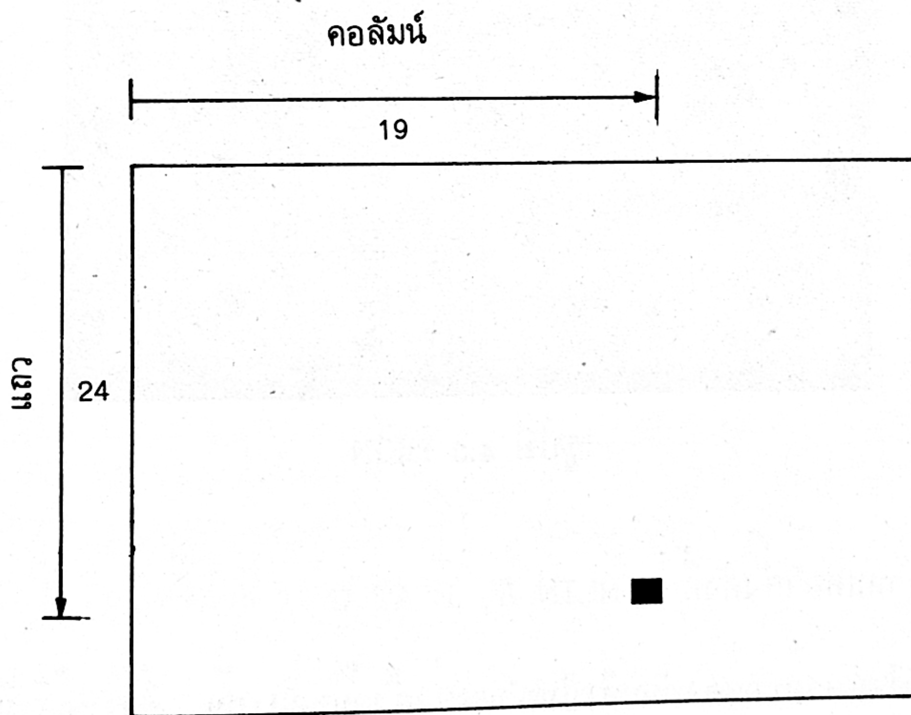
```
10   GR
20   COLOR  =  8
```

ตามตัวอย่างนี้สีจะคงอยู่เป็นสี 8 จนกว่าจะมีการเปลี่ยนสีเป็นสีอื่น

4.2.3 คำสั่งที่จะให้เขียนอะไรลงไปบนจอภาพใน LO-RES GRAPHICS MODE มีอยู่ 3 คำสั่งคือ

PLOT คือคำสั่งที่สั่งให้จอภาพแสดงภาพเป็นจุดตามตำแหน่งที่กำหนด และด้วยสีที่กำหนด ดูโปรแกรมในรูปที่ 4.4

```
10  GR               - คำสั่งที่ให้เข้า LO-RES GRAPHICS
20  COLOR = 9        - ให้สิ่งที่ปรากฏบนจอเป็นสีส้ม
30  PLOT  24, 19     - ให้แสดงภาพเป็นจุดที่แถวที่ 24 คอ-
                       ลัมน์ที่ 19
```

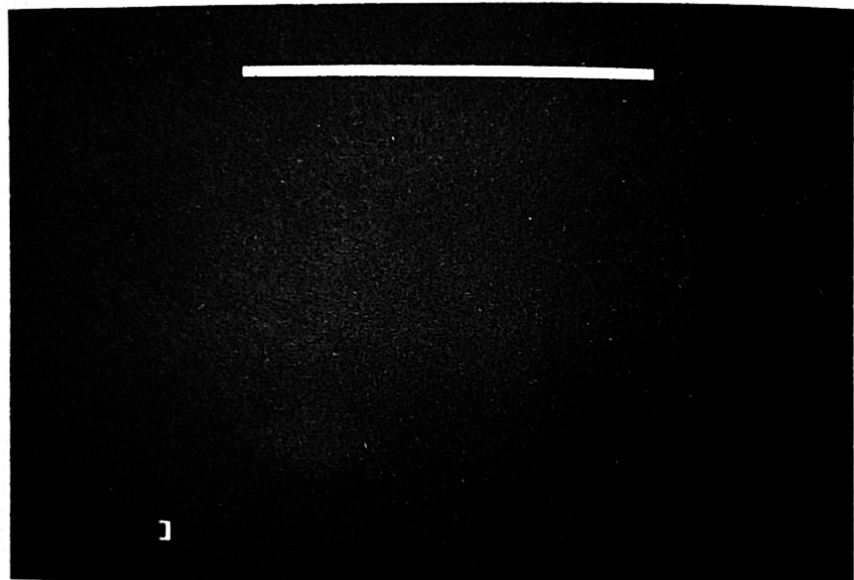


รูปที่ 4.4 คำสั่ง PLOT

พิมพ์ RUN จะได้ภาพดังแสดงในรูปที่ 4.4 เมื่อดูภาพจนพอใจแล้วให้พิมพ์ TEXT [RETURN] และ HOME [RETURN] เพื่อกลับเข้า TEXT MODE และโปรดทำดังนี้ทุกครั้งที่จะเขียนโปรแกรมใหม่

HLIN ย่อมาจาก HORIZONTAL LINE หรือเส้นระดับ คำสั่งนี้จะสั่งให้ลากเส้นตามแนวระดับ ตามตำแหน่งที่จะกำหนดด้วยตัวตามท้าย ดูโปรแกรมในรูปที่ 4.5

```
10  GR
20  COLOR = 5
30  HLIN  5, 26 AT 6
```



รูปที่ 4.5 HLIN

ถ้ากำหนดคำสั่งดังนี้ HLIN A, B AT C

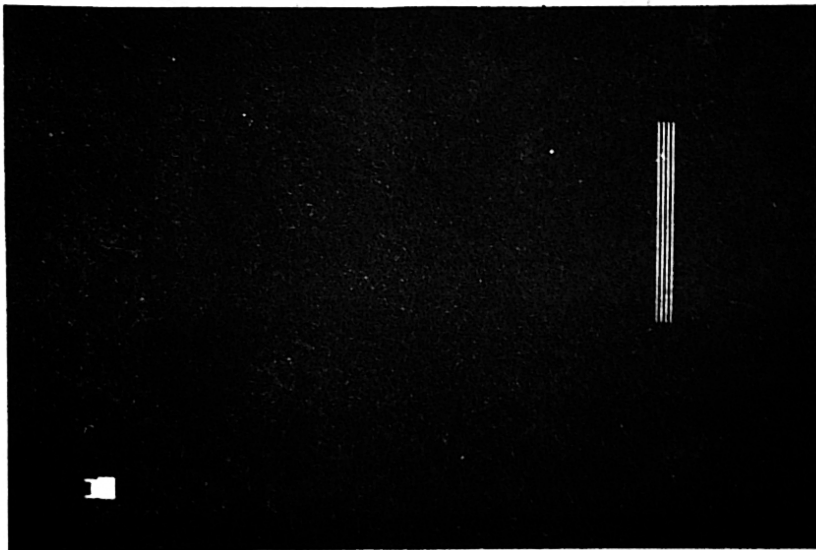
A คือหมายเลขคอลัมน์ที่เป็นตำแหน่งตั้งต้นของเส้น ซึ่งจะต้องมีค่าไม่น้อยกว่า 0

B คือหมายเลขคอลัมน์ที่เป็นตำแหน่งปลายของเส้น ซึ่งจะต้องมีค่าไม่น้อยกว่า A และไม่เกิน 39

C คือหมายเลขแถวที่เป็นตำแหน่งของเส้นระดับมีค่าอยู่ระหว่าง 0 ถึง 39

VLIN ย่อมาจาก VERTICAL LINE หรือเส้นดิ่ง คำสั่งนี้จะสั่งให้ลากเส้นตามแนวดิ่ง ตามตำแหน่งที่จะกำหนดด้วยตัวตามท้าย ดูโปรแกรมในรูปที่ 4.6

```
10  GR
20  COLOR = 8
30  VLIN  12, 30 AT 33
```



รูปที่ 4.6 VLIN

ถ้ากำหนดคำสั่งดังนี้ VLIN A, B AT C

A คือหมายเลขแถวที่เป็นตำแหน่งตั้งต้นของเส้นซึ่งจะต้องมีค่าไม่น้อยกว่า 0

B คือหมายเลขแถวที่เป็นตำแหน่งปลายของเส้นดิ่ง ซึ่งจะต้องมีค่าไม่น้อยกว่า A และไม่เกิน 35

C คือหมายเลขคอลัมน์ที่เป็นตำแหน่งของเส้นดิ่ง มีค่าอยู่ระหว่าง 0 ถึง 39 ดูโปรแกรมในรูปที่ 4.7

```
10   REM   PRO# 4.1- GRAPHICS DEMO

20   HOME
30   GR : COLOR= 15
40   SPEED= 50: PRINT "THIS IS THE
       CENTER POINT": SPEED= 255
50   PLOT 20,20
60   GOSUB 1000
70   SPEED= 50: PRINT "THIS IS THE
       TOP HORIZOTAL LINE": SPEED=
      255
80   HLIN 0,39 AT 0
90   GOSUB 1000
100   SPEED= 50: PRINT "THIS THE B
      OTTOM HORIZONTAL LINE": SPEED=
      255
110   HLIN 0,39 AT 39
120   GOSUB 1000
130   SPEED= 50: PRINT "THIS IS TH
      E LEFT VERTICAL LINE": SPEED=
      255
140   VLIN 0,39 AT 0
150   GOSUB 1000
160   SPEED= 50: PRINT "THIS IS TH
      E RIGHT VERTICAL LINE": SPEED=
      255
170   VLIN 0,39 AT 39
180   FOR I = 1 TO 5: GOSUB 1000: NEXT

190   TEXT : HOME : END
1000   FOR I = 1 TO 3000: NEXT
1010   PRINT : PRINT : PRINT : RETURN
```

รูปที่ 4.7

4.2.4 ตัวที่ตามหลังคำสั่ง PLOT, HLIN, VLIN ไม่จำเป็นที่จะต้องเป็นค่าคงที่(เป็นตัวเลข) เสมอไป อาจจะเป็นตัวแปรที่บรรจุตัวเลข (NUMERICAL VARIABLE) ก็ได้ ดูในโปรแกรมในรูปที่ 4.8

```
10   REM  PRO#4.2-GRAPHICS DEMO
20   HOME
30   GR : COLOR= 15
40   INPUT "ENTER THE 2 SIDES OF A
       RECTANGLE    <NOT OVER 40>
       ";A,B
45   PRINT : IF A > 40 OR B > 40 THEN
       FLASH : PRINT "NOT OVER 40,
       PLEASE": NORMAL : GOTO 40
50   PRINT " THE AREA OF A RECTANG
     LE WITH SIDES"
60   PRINT A;" AND ";B;" IS ";A *
     B
80   HLIN 0,A - 1 AT 0: GOSUB 1000

90   VLIN 0,B - 1 AT A - 1: GOSUB
     1000
100  HLIN A - 1,0 AT B - 1: GOSUB
     1000
110  VLIN B - 1,0 AT 0: GOSUB 100
     0
120  FOR I = 1 TO 9: GOSUB 1000: NEXT

130  TEXT : HOME
140  HTAB 12: VTAB 13: PRINT "ANY
      MORE ? <Y/N>"
150  INPUT A$
160  IF A$ = "Y" THEN 20
170  END
1000  FOR I = 1 TO 2000: NEXT : RETURN
```

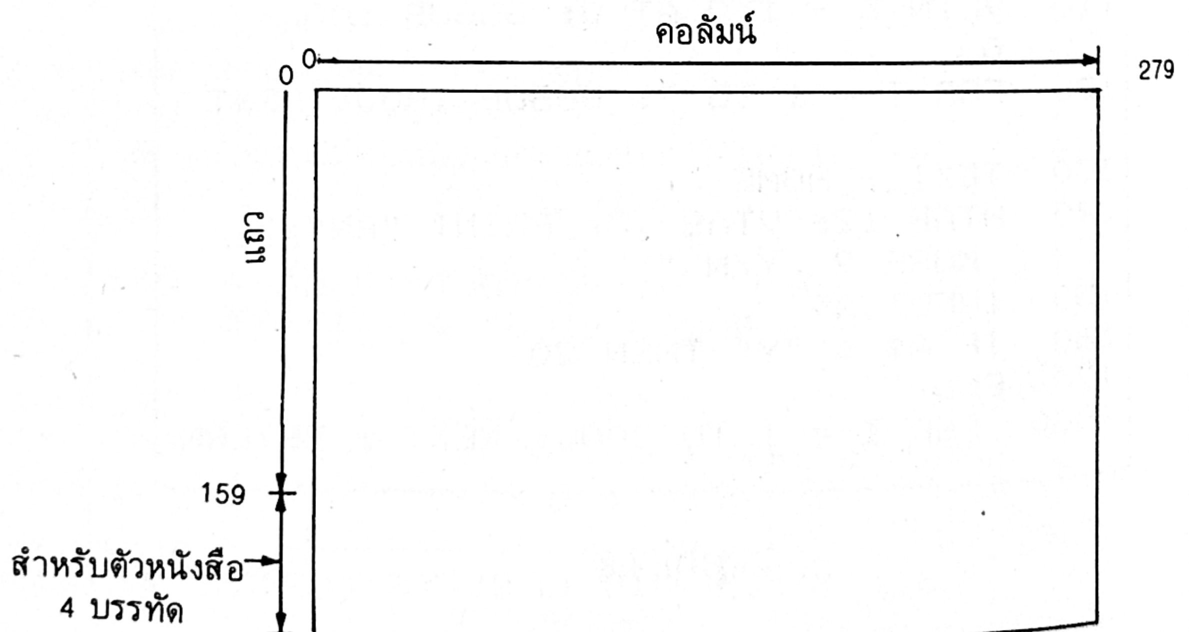
รูปที่ 4.8

4.3 กราฟิกเส้นละเอียด (HIGH RESOLUTION GRAPHICS)

หรือที่เรียกย่อๆ ว่า HI-RES GRAPHICS ซึ่งจะแบ่งออกเป็น 2 แบบเช่นกันคือ แบบ SPLIT SCREEN (HGR) และแบบ FULL SCREEN (HGR2)

4.3.1 เราจะเข้าสู่ HI-RES SPLIT SCREEN GRAPHICS MODE ด้วยการพิมพ์ HGR ซึ่งหลังจากนี้จอภาพส่วนบนจะถูกแบ่งออกเป็นตาราง(ซึ่งมองไม่เห็น) จากซ้ายไปขวาจะแบ่งเป็น 280 คอลัมน์ (คอลัมน์ที่ 0 ถึงคอลัมน์ที่ 279) และจากบนลงมาล่างจะแบ่งเป็น 160 แถว(แถวที่ 0 ถึงแถวที่ 159) ด้านล่างของจอจะถูกจัดให้เป็นช่องสำหรับเขียนอักษรได้ 4 บรรทัด (ดูรูปที่ 4.9)

เมื่อพิมพ์ HGR ส่วนบนของจอภาพ นอกจากจะถูกแบ่งออกเป็นตาราง (ซึ่งมองไม่เห็น) ดังกล่าวแล้ว จอภาพส่วนนั้นจะเปลี่ยนเป็นสีดำ เครื่องหมาย PROMPT กับ CURSOR ถ้าอยู่ส่วนบนของจอ(หลังจาก HOME)จะหายไปด้วย นอกจากจะพิมพ์ VTAB 21 จึงจะมาปรากฏที่ส่วนล่างของจอ



รูปที่ 4.9 HGR

หากคำสั่งที่พิมพ์เป็น HGR2 จอภาพทั้งหมดจะแบ่งเป็นตาราง จากซ้ายไปขวาจะแบ่งเป็น 280 คอลัมน์(คอลัมน์ที่ 0 ถึงคอลัมน์ที่ 279) จากบนลงมาล่างจะแบ่งเป็น 192 แถว(แถวที่ 0 ถึงแถวที่ 191) และไม่มี TEXT WINDOW อยู่ในจอภาพ

4.3.1 ใน LO-RES GRAPHICS เราสามารถจะให้สีปรากฏลงบนจอได้ถึง 16 สี(ในจอภาพที่เป็นชนิดที่มีสีหรือ T.V. สี) ใน HI-RES GRAPHICS เราสามารถบรรจุสีได้เพียง 8 สีเท่านั้น และใน 8 สีนั้นจะแบ่งเป็นสีคู่ และสีคี่ดังแสดงในรูปที่ 4.10 ดังนี้

สีคู่	สีคี่
0 ดำ 1	1 เขียว (ขึ้นอยู่กับชนิดของจอ)
2 น้ำเงิน (ขึ้นอยู่กับชนิดของจอ)	3 ขาว 1
4 ดำ 2	5 (ขึ้นอยู่กับชนิดของจอ)
6 (ขึ้นอยู่กับชนิดของจอ)	7 ขาว 2

รูปที่ 4.10 สีใน HI-RES GRAPHICS

สีคู่จะปรากฏอยู่บนแถวที่มีหมายเลขคู่และสีคี่จะปรากฏอยู่บนแถวที่มีหมายเลขคี่เท่านั้น สำหรับจอขาวดำแล้วจะไม่ปรากฏสีแต่จะมีความแตกต่างกันให้เห็น

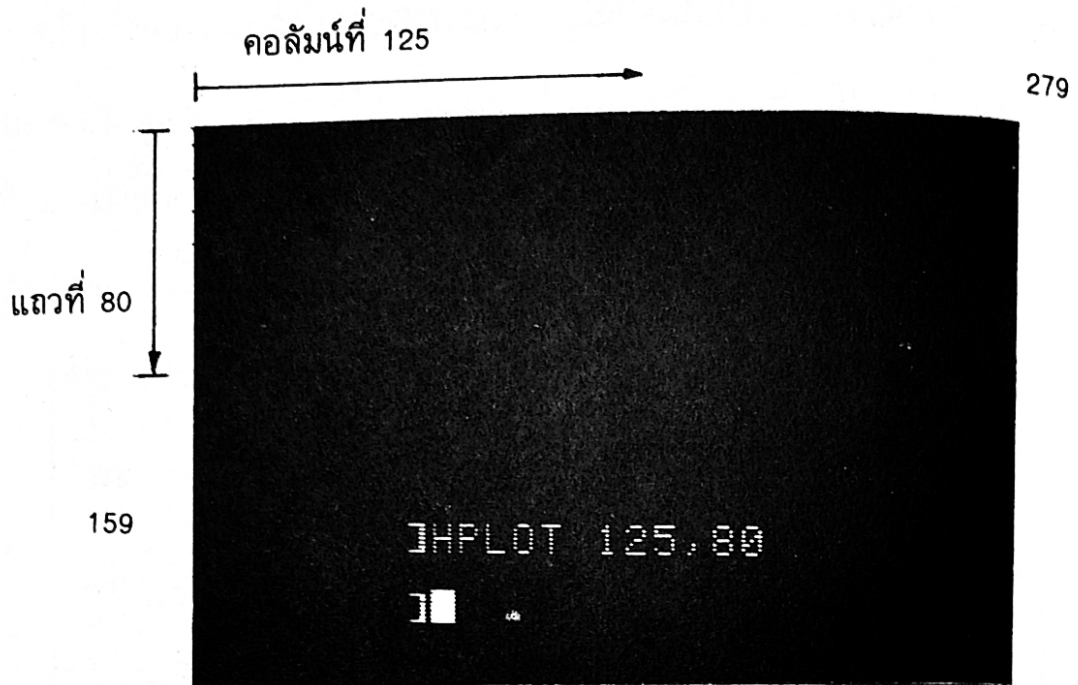
การกำหนดสีให้กด HCOLOR ตามด้วยเครื่องหมายเท่ากับและหมายเลขสี เช่น

HCOLOR = 3

4.3.2 คำสั่งให้ PLOT ใน HI-RES GRAPHICS ทำให้กว้างขวางและสนุกสนานกว่าใน LO-RES GRAPHICS มาก

คำสั่งให้ PLOT ใน HI-RES คือ HPLOT มี 3 แบบคือ

(1) HPLOT 125, 80 เป็นการสั่งให้แสดงจุดในคอลัมน์ที่ 125 ตัดกับแถวที่ 80 ดูรูปที่ 4.11



รูปที่ 4.11 HPLOT 125, 80

หมายเหตุ ก่อนที่จะเขียนคำสั่งนี้จะต้องมีคำสั่งนำดังนี้

VTAB 21

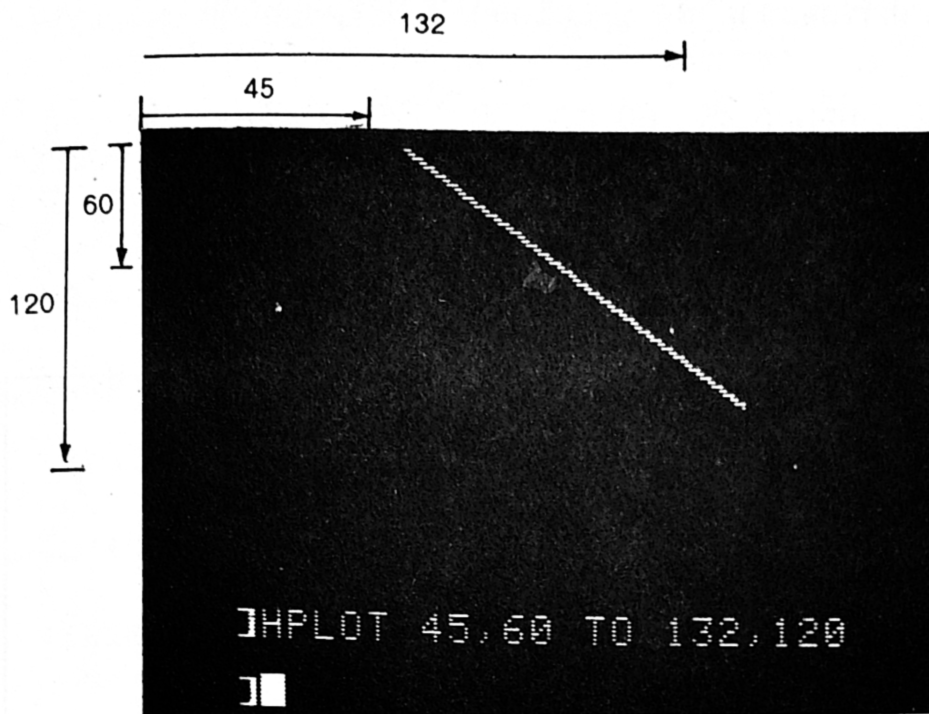
HGR

COLOR = 3

(โปรดสังเกตว่าเราใช้ IMMEDIATE EXECUTION COMMAND)

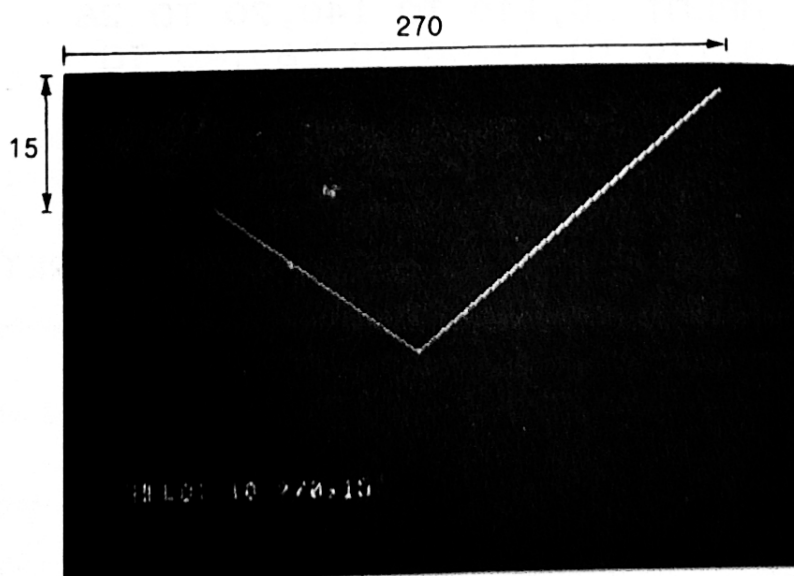
คำสั่ง VTAB 21 เป็นการย้าย CURSOR ลงมาบรรทัดที่ 21 มิฉะนั้นเมื่อเราออกคำสั่ง HGR แล้ว บนจอภาพตั้งแต่บรรทัดที่ 21 ขึ้นไปจะเปลี่ยนเป็นสีดำ คำสั่งต่อไปที่เราพิมพ์จะไม่ปรากฏอยู่บนจอ ทำให้ไม่ทราบว่าพิมพ์ผิดหรือถูก

(2) HPLOT 45, 60 TO 132, 120 เป็นการสั่งให้ลากเส้นจากจุดที่คอลัมน์ 45 ตัดกับแถวที่ 60 มายังจุดที่คอลัมน์ 132 ตัดกับแถวที่ 120 ดูรูปที่ 4.12



รูปที่ 4.12 HPLOT 45, 60, TO 132, 120

(3) HPLOT TO 270, 15 คำสั่งนี้จะต้องต่อเนื่องมาจากคำสั่ง HPLOT อื่นๆ จะเป็นการสั่งให้ PLOT จากจุดสุดท้ายมายังจุดใหม่ ในรูปที่ 4.11 ถ้าเขียนคำสั่งดังกล่าวลงไป ภาพบนจอจะลากเส้นเพิ่มเติมจากรูปเดิม ดูรูปที่ 4.13



รูปที่ 4.13

เราสามารถจะนำคำสั่ง HPLOT ทั้ง 2 มารวมกันดังนี้

HPLOT 45, 60 TO 132, 120 TO 270, 15

สังเกตดูโปรแกรมในรูปที่ 4.14

```
10   REM  PRO#4.3
20   HOME : VTAB 8: HTAB 10: PRINT
      "HI-RES GRAPHICS DEMO"
30   GOSUB 1000: HOME : HGR : VTAB
      21
40   PRINT "HERE'S THE TEXT WINDOW
      "
50   HCOLOR= 7
60   HPLOT 20,140 TO 140,20: GOSUB
      1000
70   HPLOT  TO 260,140: GOSUB 1000

80   HPLOT  TO 20,40: GOSUB 1000
90   HPLOT  TO 260,40: GOSUB 1000
100  HPLOT  TO 20,140: GOSUB 1000

110  FOR I = 1 TO 5: GOSUB 1000: NEXT

120  TEXT : HOME : HGR : HCOLOR=
     7: GOSUB 1000
130  HPLOT 20,140 TO 140,20 TO 26
     0,140 TO 20,40 TO 260,40 TO
     20,140
140  FOR I = 1 TO 10: GOSUB 1000:
      NEXT
150  TEXT : HOME : END
1000  FOR I = 1 TO 2000: NEXT I: RETURN
```

รูปที่ 4.14

ในโปรแกรมนี้จะมีข้อสังเกตอยู่ 3 ประการคือ

1. การจะเขียนกราฟิกแบบเส้นละเอียดจะต้องเริ่มด้วยคำสั่ง HGR (บรรทัดที่ 30) และ HCOLOR = ... (บรรทัดที่ 50) เสมอไป
2. ถ้าไม่มีคำสั่ง VTAB 21 ในบรรทัดที่ 30 แล้ว การพิมพ์ในบรรทัดที่ 40 จะไม่ปรากฏ เพราะเมื่อเครื่องได้รับคำสั่งแล้ว บริเวณส่วนบนของจอจะเป็นสีดำอยู่
3. โปรดสังเกตว่าเราจะ PLOT เส้นทีละเส้นหรือจะ PLOT ทีเดียว โดยเขียนคำสั่งต่อเนื่องกันไปเลยก็ได้

```
10   REM   PRO#4.4
20   HOME : HGR : VTAB 21
30   FOR J = 7 TO 0 STEP  - 1
40   HCOLOR= J
50   PRINT "HCOLOR = ";J: PRINT : PRINT

60   FOR I = 0 TO 159
70   HPLOT 0,I TO 270,I
80   NEXT I
90   NEXT J
100   TEXT : HOME : END
```

รูปที่ 4.15

รูปที่ 4.15 จะเป็นโปรแกรมที่แสดงการลากเส้นระดับไล่ลงมา ตั้งแต่แถวที่ 0 จนถึงแถวที่ 159 และแสดงความแตกต่างของสีตั้งแต่สีที่ 7 ถึงสีที่ 0 ลองตัดคำสั่ง PRINT 2 ตัวในบรรทัดที่ 50 ออกลอง RUN ดู แล้วลองตัดคำสั่ง VTAB 21 ในบรรทัดที่ 20 ลอง RUN แล้วลองพิจารณาหาเหตุผลว่าเหตุใดจึงไม่แสดงผลดีเท่าที่ควร

```
10  REM  PRO#4.5
20  HOME : DIM X(30),Y(30)
30  HGR2 : HCOLOR= 7
40 N =  INT ( RND (1) * 14 + 7)
50 PI = 2 * 3.14159 / N
60  FOR I = 0 TO N - 1
70 Z = PI * I
80 X(I) =  COS (Z) * 107 + 140
90 Y(I) =  SIN (Z) * 95 + 96
100  NEXT I
110  FOR I = 0 TO N - 1
120  FOR M = I TO N - 1
130  HPLOT X(I),Y(I) TO X(M),Y(M)

140  NEXT M,I
150  FOR I = 1 TO 1000: NEXT : GOTO
     30
```

รูปที่ 4.16 HGR 2

รูปที่ 4.16 เป็นการ PLOT รูปเหลี่ยมต่างๆ ที่ให้ผลออกมาอย่างสวยงาม-มาก

สรุป

1. กราฟิกมีอยู่ 2 แบบคือ กราฟิกเส้นหยาบ(LO-RES GRAPHICS) ซึ่งมีทั้ง SPLIT SCREEN ซึ่งมี TEXT WINDOW อยู่ส่วนล่างของจอภาพและ FULL SCREEN ซึ่งเป็นกราฟิกเต็มทั้งจอกับกราฟิกเส้นละเอียด(HI-RES GRAPHICS) ซึ่งมีทั้ง SPLIT SCREEN (HGR) และ FULL SCREEN (HGR2)

2. ตารางในรูปที่ 4.14 จะแสดงจำนวนตารางของจอภาพที่ถูกแบ่งออกเนื่องจากสั่งให้เครื่องทำงานใน GRAPHICS MODE

LO-RES GRAPHICS		HI-RES GRAPHICS	
SPLIT SCREEN	FULL SCREEN	HGR	HGR2
40 × 40	40 × 48	280 × 160	280 × 192

รูปที่ 4.17 แสดงการแบ่งตารางบนจอภาพ

3. คำสั่งใน LO-RES GRAPHICS มีดังนี้ GR, COLOR =, HLIN, VLIN
4. คำสั่งใน HI-RES GRAPHICS มีดังนี้ HGR, HGR2, HCOLOR =, HPLOT

แบบฝึกหัดบทที่ 4

4.1 ในหนังสือ "เรียน APPLESOFT BASIC ด้วยตนเอง" หน้า 110 มีตัวอย่างการใช้ HTAB, VTAB เขียนโปรแกรมให้แสดงคำว่า "สวัสดี" ลงบนจอ จงเขียนคำนี้โดยใช้ LO-RES GRAPHICS

4.2 จง PLOT กราฟของสมการ R = A COS 2I ให้ I = 0 ถึง 2π, A = 50

เฉลยแบบฝึกหัดบทที่ 4

4.1

```
10 REM SOL#4.1
20  HOME : GR : COLOR= 15
30  HLIN 7,12 AT 6: HLIN 14,19 AT
     6: HLIN 21,26 AT 6: HLIN 28,
     33 AT 6
40  HLIN 7,12 AT 9: HLIN 21,26 AT
     9
50  HLIN 7,9 AT 13: HLIN 17,19 AT
     13: HLIN 21,23 AT 13: HLIN 2
     8,30 AT 13
60  HLIN 7,12 AT 9: HLIN 21,26 AT
     9
70  VLIN 10,12 AT 7: VLIN 7,13 AT
     12: VLIN 7,12 AT 19
80  VLIN 10,12 AT 21: VLIN 7,13 AT
     26: VLIN 7,12 AT 28: VLIN 7,
     13 AT 33
90  VLIN 4,5 AT 10: VLIN 4,5 AT 2
     4
100  PLOT 14,3: HLIN 14,19 AT 4: HLIN
      28,33 AT 4: PLOT 33,3
110  FOR I = 1 TO 2000: NEXT
120  TEXT : HOME
```

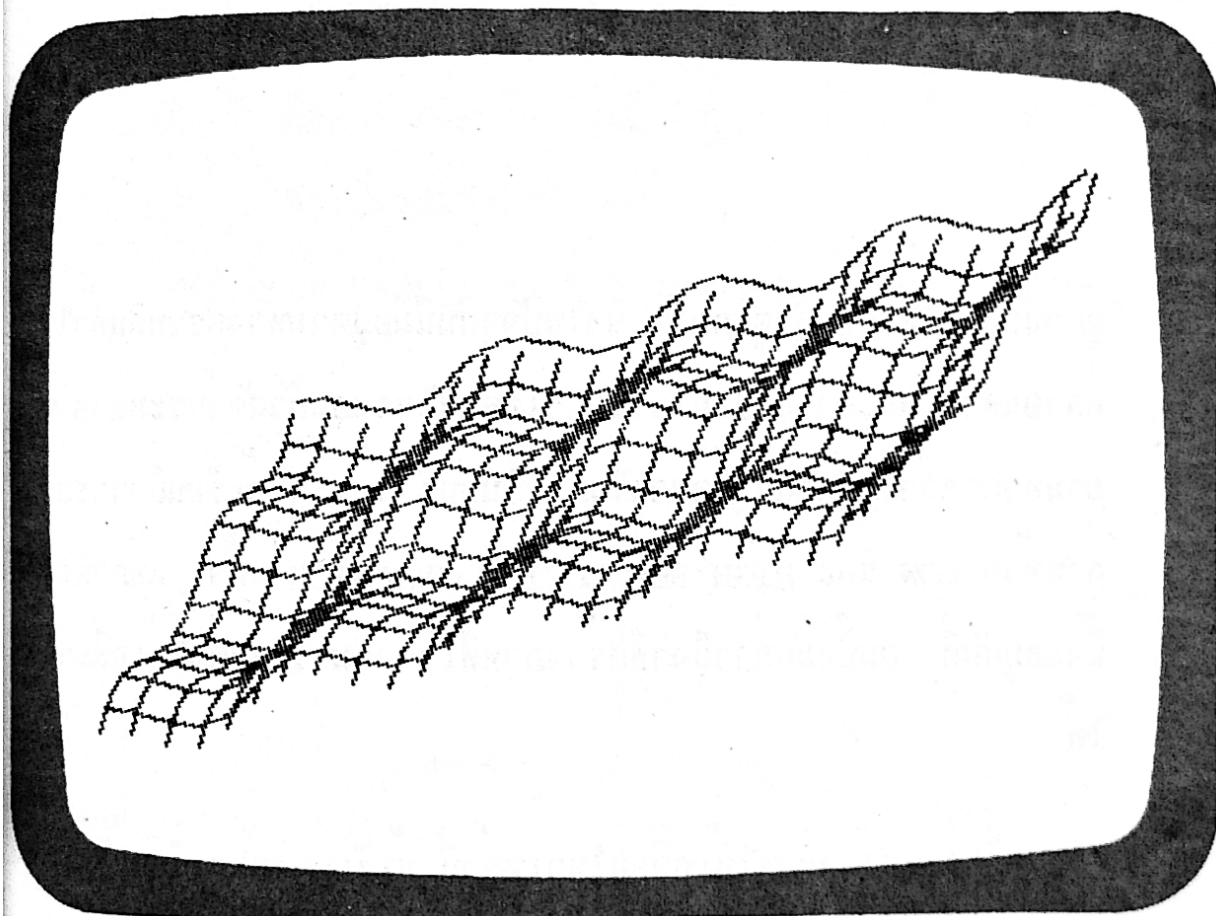
ลองปรับปรุงโปรแกรมให้ดีขึ้นโดยใช้ READ...DATA

```
10   REM  SOL#4.1MOD
20   HOME : GR : COLOR= 15
30   FOR I = 1 TO 14
40   READ A,B,C
50   HLIN A,B AT C
60   NEXT
70   FOR I = 1 TO 9
80   READ D,E,F
90   VLIN D,E AT F
100   NEXT
110   PLOT 14,3: PLOT 33,3
```

```
115  FOR I = 1 TO 3000: NEXT : TEXT
     : HOME
120  DATA  7,12,6,14,19,6,21,26,6
     ,28,33,6,7,12,9,21,26,9,7,9,
     13,17,19,13
130  DATA  21,23,13,28,30,13,7,12
     ,9,21,26,9,14,19,4,28,33,4
140  DATA   10,12,7,7,13,12,7,12,
     19,10,12,21,7,13,26,7,12,28,
     7,13,33,4,5,10,4,5,24
```

4.2

```
10  REM  SOL#4.2
15 A1 = 0:A2 = 2 * PI
20  HGR : HCOLOR= 3:PI = 3.14159
30 A1 = 0:A2 = 2 * PI
40 N = 90:A = 50
50 INC = (A2 - A1) / N
60  FOR I = A1 TO A2 STEP INC
70 R = A *  COS (2 * I)
80 X = R *  SIN (I)
90 Y = R *  COS (I)
100  HPLOT 140 + X,80 + Y
110  NEXT I
120  FOR I = 1 TO 3000: NEXT I: TEXT
     : HOME
```



5

SHAPE TABLE

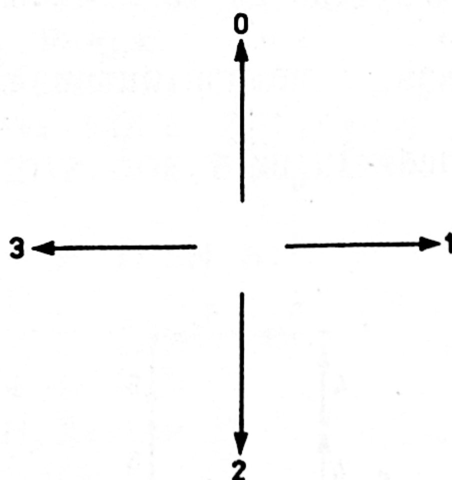
5.1 ผู้อ่านคงจะได้เคยเห็นตู้เกม ซึ่งในปัจจุบันนี้มีอยู่ตามห้างสรรพสินค้าใหญ่ๆ หลายแห่ง และคงจะแปลกใจในการเคลื่อนไหวของตัวสัตว์ประหลาด หรือยานอวกาศว่าเขาทำกันได้อย่างไร ในบทที่ 4 เราได้เรียนถึงการสร้างภาพด้วย LOW และ HIGH RESOLUTION GRAPHICS แล้ว แต่ภาพเหล่านี้จะอยู่กับที่ บทนี้จะกล่าวถึงวิธีสร้างภาพที่เราสามารถจะสั่งให้เคลื่อนไหวได้

ความจริงการสร้างภาพให้เคลื่อนไหวเหล่านี้ ผู้เรียนจะต้องเรียนรู้ให้ลึกซึ้งถึงโครงสร้างหน่วยความจำของเครื่องคอมพิวเตอร์ ซึ่งจะหาอ่านละเอียดได้ในหนังสือ "APPLESOFT BASIC REFERENCE MANUAL" แต่เรื่องนี้ค่อนข้างจะยากมากสำหรับผู้เริ่มเล่น ซึ่งยังไม่มีพื้นฐานในเรื่องเลขฐานสอง (BINARY) และเลขฐานสิบหก(HEXADECIMAL) ในหนังสือนี้จะกล่าวเพียงแต่ว่าเราจะสร้างรูปต่างๆ ได้อย่างไรโดยใช้โปรแกรมสำเร็จรูป ผู้สนใจเพิ่มเติมคงจะศึกษาต่อไปได้ในหนังสือดังกล่าวและเล่มอื่นๆ

5.2 การเคลื่อนไหวของ *รูปร่าง (SHAPE)* นั้น จะถือเกณฑ์จุดศูนย์กลางของรูปเป็นหลัก ดังนั้นในการสร้างรูปร่างจึงต้องมีการกำหนดจุดศูนย์กลางของรูป และลากเส้นออกจากจุดศูนย์กลางดังกล่าวไปตามเส้นรอบรูปของรูปร่างนั้นจนมาบรรจบที่เดิม ในการลากเส้นดังกล่าวนี้จะมีอยู่ 2 แบบคือ แบบหนึ่งลากเส้นไม่ให้มีเส้นปรากฏอยู่บนจอ อีกแบบหนึ่งให้เส้นที่ลากไปนั้นปรากฏอยู่บนจอด้วย ในที่นี้จะขอใช้คำต่อไปนี้แทน

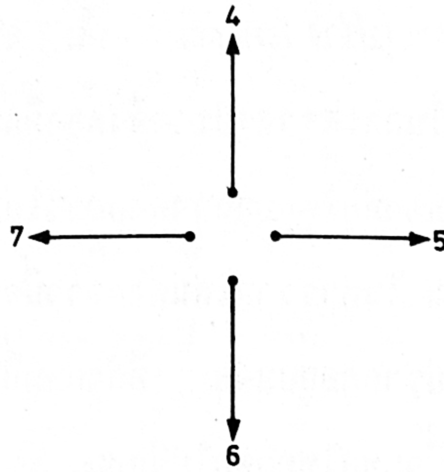
เคลื่อนที่ คือการเคลื่อนที่จากจุดถึงจุด โดยการเคลื่อนที่นั้นไม่ปรากฏบนจอเป็นรูปร่าง

ลากเส้น คือการลากเส้นจากจุดถึงจุด โดยให้เส้นที่ลากนั้นปรากฏเป็นรูปร่าง



รูปที่ 5.1 รหัสและทิศทางของการเคลื่อนที่

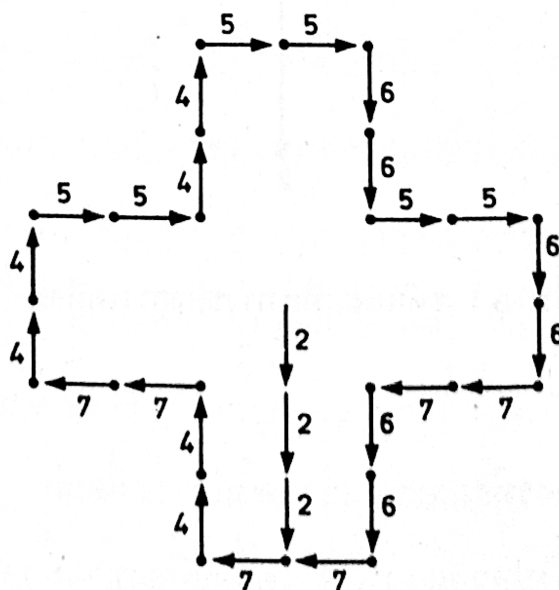
รูปที่ 5.1 แสดงรหัสและทิศทางของการเคลื่อนที่ กล่าวคือ 0 เคลื่อนที่ขึ้น, 1 เคลื่อนที่ไปทางขวา, 2 เคลื่อนที่ลงข้างล่าง, 3 เคลื่อนที่ไปทางซ้าย



รูปที่ 5.2 รหัสและทิศทางของการลากเส้น

ในรูปที่ 5.2 รหัส 4 คือการลากเส้นขึ้นข้างบน, 5 ลากเส้นไปทางขวา, 6 ลากเส้นไปข้างล่าง และ 7 ลากเส้นไปทางซ้าย

สมมติว่าจะสร้างรูปร่างเป็นรูปเครื่องหมายกากบาท เราจะเริ่มต้นที่จุดศูนย์กลางเคลื่อนที่ลงข้างล่าง(รหัส 2) ไป 3 หน่วย แล้วลากเส้นมาทางซ้าย (รหัส 7) ไป 1 หน่วย แล้วลากเส้นขึ้นข้างบน(รหัส 4) ไป 2 หน่วย ดังนี้จนครบเป็นรูปดังแสดงในรูปที่ 5.3



รูปที่ 5.3 การสร้างรูป

เรานำรหัสดังกล่าวนี้มาเขียนเรียงกันได้ดังนี้

2, 2, 2, 7, 4, 4, 7, 7, 4, 4, 5, 5, 4, 4, 5, 5, 6, 6, 5, 5, 6, 6, 7, 7, 6, 6, 7

จากนี้เราจะ SAVE ตัวเลขเหล่านี้เข้าไว้หน่วยความจำของคอมพิวเตอร์ซึ่งเราเรียกว่าเป็น *SHAPE TABLE* หรือ *ตารางรูปร่าง*

รหัสดังกล่าวนี้จะต้องแปลงเป็นเลขฐานสิบหก และ SAVE ไว้ในหน่วยความจำที่กำหนดเป็นพิเศษ ซึ่งจะไม่กล่าวถึงในที่นี้ แต่จะให้ไว้ในรูปของ SUBROUTINE ดังแสดงในรูปที่ 5.4

```
500  REM  PRO#5.1 SUBROUTINE FOR
     SHAPE TABLE
510  POKE 768,01: POKE 769,00: POKE
     770,04: POKE 771,00:M = 772
520  POKE 232,00: POKE 233,03
530  READ A
540  IF A = 8 THEN 610
550  READ B
560  IF B = 8 THEN 600
570 X = B * 8 + A
580  POKE M,X:M = M + 1
590  GOTO 530
600  POKE M,A:M = M + 1
610  POKE M,0
620  RETURN
1000  DATA  2,2,2,7,4,4,7,7,4,4,5
     ,5,4,4,5,5,6,6,5,5,6,6,7,7,6
     ,6,7,8
```

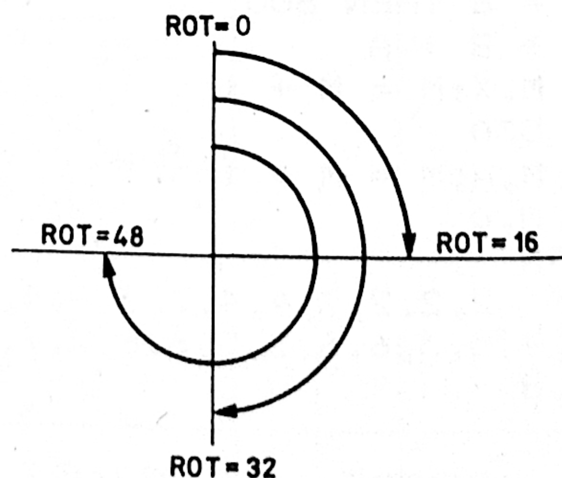
รูปที่ 5.4 SUBROUTINE ของ SHAPE TABLE

SUBROUTINE ในรูปที่ 5.4 นี้ เป็นการเก็บ SHAPE TABLE ไว้ในหน่วยความจำของคอมพิวเตอร์ บรรทัดที่ 1000 คือรหัสที่เป็นรูปร่างของกากบาท โปรดสังเกตว่าจะต้องมีรหัส 8 ปิดท้าย แสดงว่าหมดรูปร่างนั้น

5.3 หลังจากที่ได้สร้างรูปร่างและเก็บตารางรูปร่างไว้ในหน่วยความจำแล้วเราจะนำรูปร่างนั้นมาแสดงใน HI-RES GRAPHICS MODE ด้วย คำสั่ง HGR และคำสั่งอื่นที่จะนำมาใช้ในการแสดงรูปร่างนั้นมีดังนี้

SCALE อัตราส่วนที่จะให้นำมาแสดงบนจอ SCALE = 1 รูปร่างที่ออกมาจะเท่ากับที่เขียนไว้ (ตามตัวอย่างรูปกากบาทจะมีปีกกว้างด้านละ 2 ช่อง HGR) SCALE = 10 รูปร่างที่จะแสดงบนจอจะใหญ่เป็น 10 เท่า

ROT ย่อมาจาก ROTATE คือคำสั่งที่จะหมุนภาพ โดยใช้จุดที่เริ่มต้นเขียนเป็นแกนหมุน ค่าของ ROT จะอยู่ระหว่าง 0 ถึง 255 ค่าของ ROT เท่ากับ 0, 16, 32 หรือ 48 รูปร่าง



รูปที่ 5.5 การหมุนรูป

จะหมุนทำมุม 0, 90, 180 หรือ 270 องศา (ตามเข็มนาฬิกา) ตามลำดับ

ในรูปที่กำหนดให้ SCALE = 1 ค่าของ ROT ที่เครื่องจะหมุนให้คือ 4 ค่าดังกล่าว ถ้าเรากำหนดค่าของ ROT อยู่ระหว่าง 2 ค่าใด(สมมติว่าให้ ROT = 20) เครื่องจะปัดไปค่าที่ตัวเลขน้อย(คือให้ ROT = 16) ถ้ากำหนดให้ SCALE = 2 ค่าที่เครื่องจะหมุนให้จะมี 8 ค่า

DRAW... มีรูปแบบของคำสั่งคือ DRAW N AT X, Y คือคำสั่งให้เขียนรูปร่างตามตารางรูปร่างในหน่วยความจำ N คือหมายเลขของรูปร่างโปรแกรมตามรูปที่ 5.4 จะเขียนได้รูปเดียว ดังนั้น N จึงเป็น 1 เสมอ X, Y คือตำแหน่งที่จุดเริ่มต้น (ปกติคือจุดศูนย์กลาง) ของรูปร่างที่จะปรากฏอยู่บนจอ

XDRAW มีรูปแบบคือ XDRAW N AT X, Y คือคำสั่งที่ให้รูปร่างที่ปรากฏอยู่ในตำแหน่งที่ระบุลบหายไปโดยที่สีพื้นยังคงปรากฏอยู่ไม่สบเลือนไป

5.4 ลองใช้คำสั่งทั้ง 3 ดังกล่าวตามลำดับดังนี้

- พิมพ์โปรแกรมตามรูปที่ 5.4 ลงไป
- พิมพ์บรรทัดต่อไปนี้ลงไป (ตามด้วย RETURN ทุกบรรทัด)

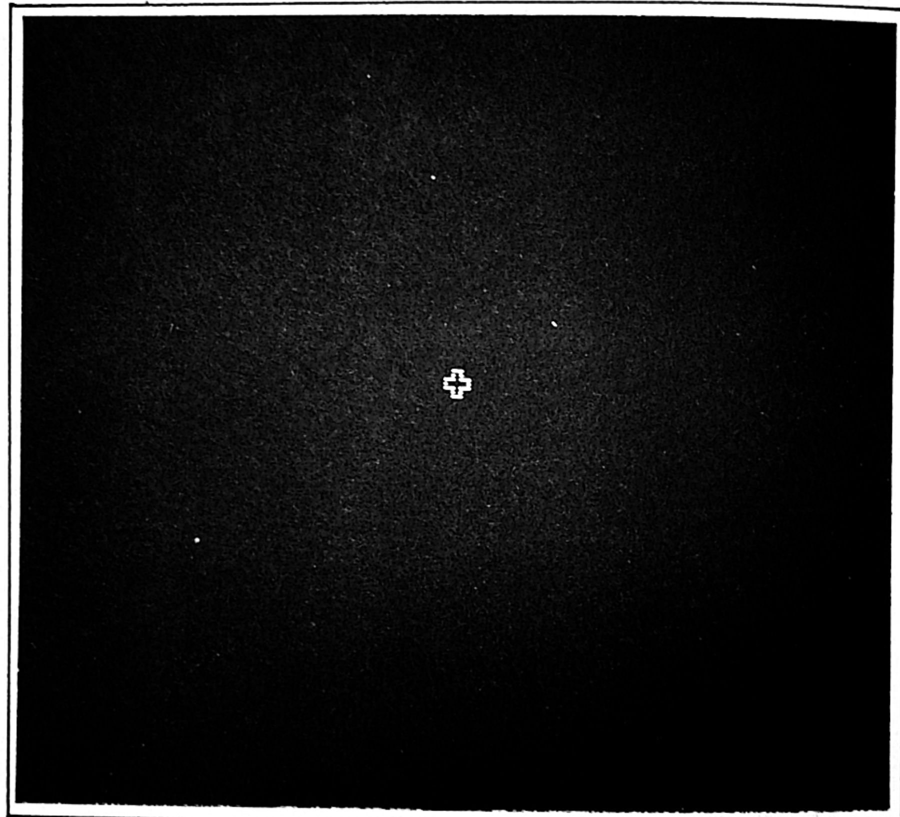
```
GOSUB 500
HGR : HCOLOR = 3
```

```
SCALE = 1

ROT = 0

DRAW 1 AT 140, 80
```

ภาพจะปรากฏดังแสดงในรูปที่ 5.6



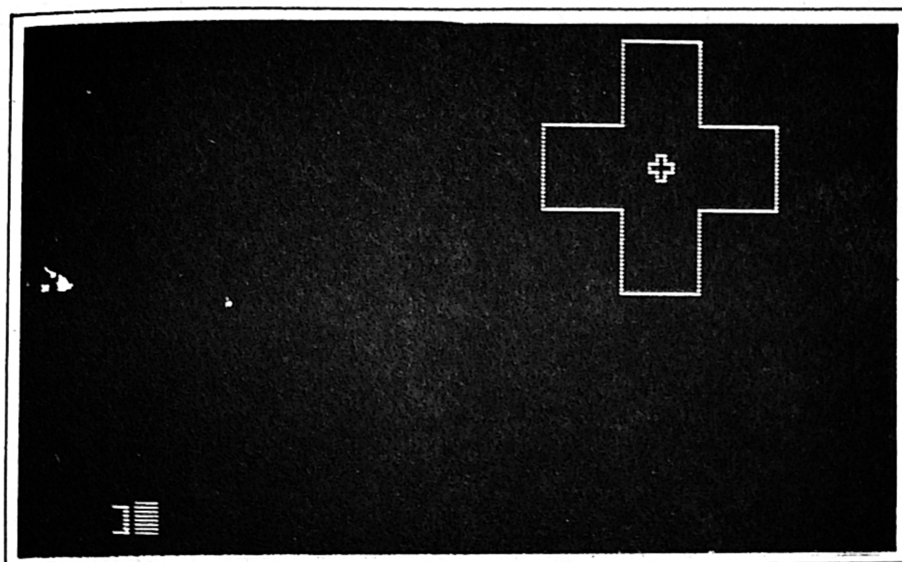
รูปที่ 5.6

ลองเปลี่ยน SCALE โดยการพิมพ์ต่อไปนี้

```
SCALE = 10

DRAW 1 AT 140, 80
```

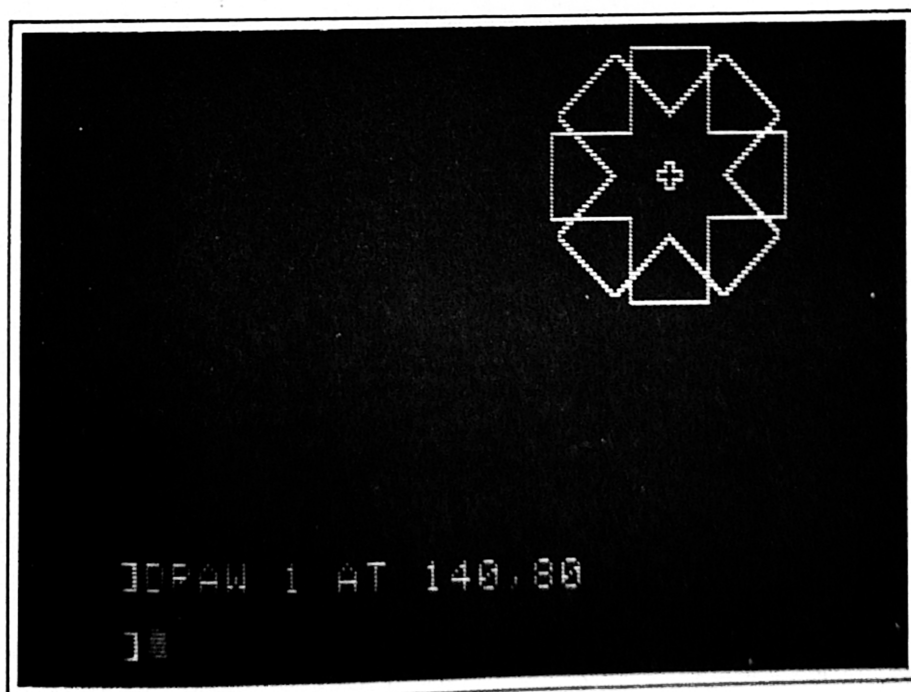
ภาพจะปรากฏดังในรูปที่ 5.7



รูปที่ 5.7

ต่อไปลองหมุนรูปไป 45 องศาด้วยการพิมพ์

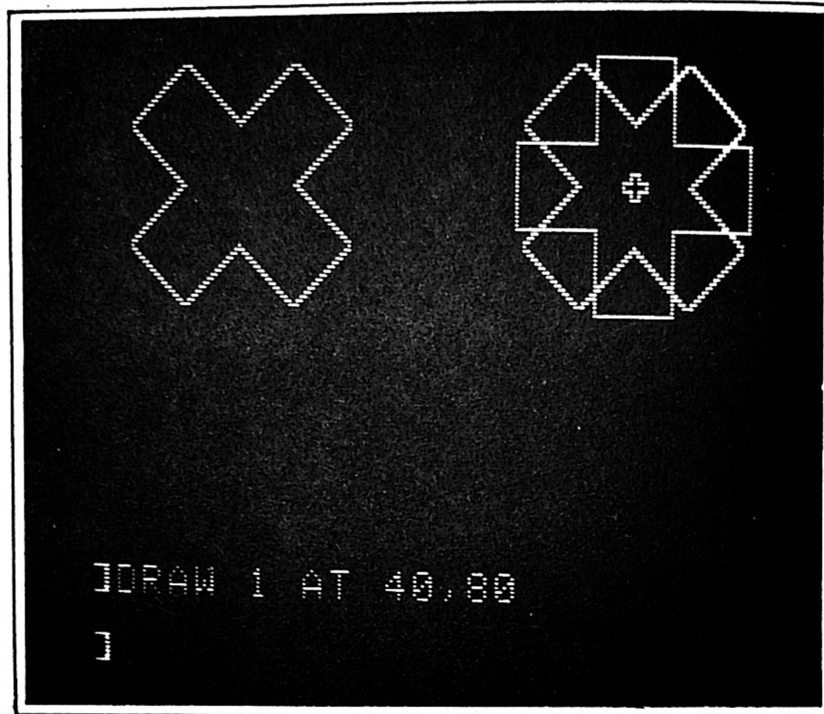
```
ROT = 8
DRAW 1 AT 140, 80
```



รูปที่ 5.8

ลองย้ายรูปด้วยการพิมพ์

DRAW 1 AT 80, 40



รูปที่ 5.9

ลองพิมพ์ XDRAW 1 AT 80, 40

ลองนำคำสั่งเหล่านี้บรรจุลงในโปรแกรม

```
10  REM  PRO#5.2
20  HOME : HGR : HCOLOR= 3
30  GOSUB 500
40  SCALE= 15: ROT= 0: DRAW 1 AT
     90,80
50  SCALE= 3: ROT= 8: DRAW 1 AT 2
     00,40
60  SCALE= 4: ROT= 4: DRAW 1 AT 2
     00,120
70  END
```

รูปที่ 5.10

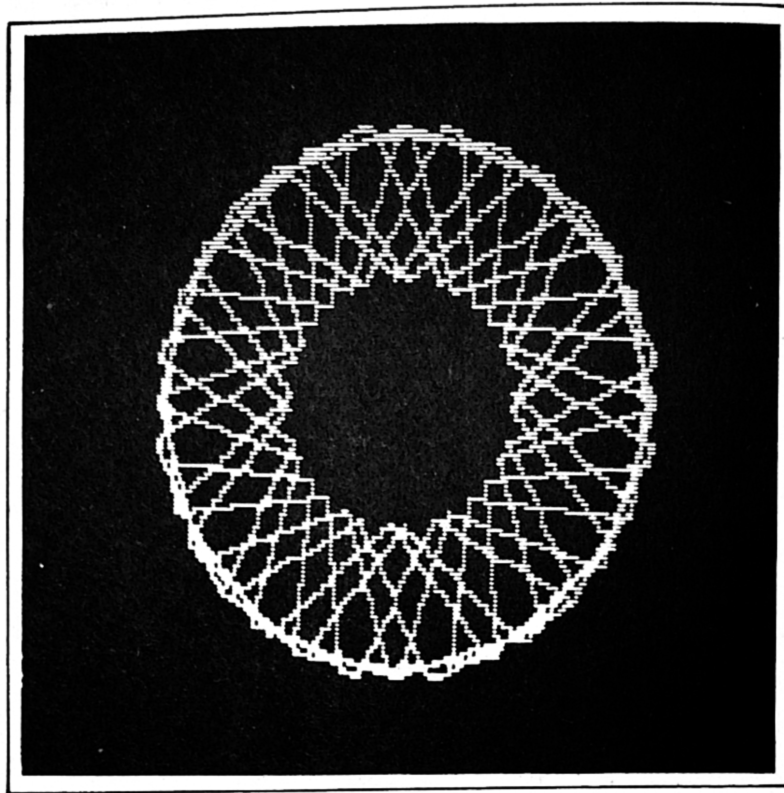
อย่าลืมว่าโปรแกรมนี้จะใช้ได้ก็ต่อเมื่อมี SUBROUTINE ในรูปที่ 5.4 ด้วย

โปรแกรมในรูปที่ 5.11 การเขียนรูปจะซ้ำกันอยู่ ลองเพิ่มบรรทัดที่ 90 ดู

```
90  XDRAW 1 AT 140,80
```

บรรทัดนี้จะลบรูปที่ได้เขียนไว้แล้ว และเมื่อรูปใหม่ถูกเขียนขึ้นมาจะไม่มีรูปซ้อนกัน

```
10   REM  PRO#5.3
20   HOME : HGR : HCOLOR= 3
30   GOSUB 500
40   FOR I = 1 TO 48 STEP 6
50   SCALE= 20
60   ROT= I
70   DRAW 1 AT 140,80
80   GOSUB 200
100  NEXT I
110  GOSUB 200: TEXT : HOME
120  END
200  FOR J = 1 TO 1000: NEXT : RETURN
500  REM  PRO#5.1 SUBROUTINE FOR
     SHAPE TABLE
510  POKE 768,01: POKE 769,00: POKE
     770,04: POKE 771,00:M = 772
520  POKE 232,00: POKE 233,03
530  READ A
540  IF A = 8 THEN 610
550  READ B
560  IF B = 8 THEN 600
570 X = B * 8 + A
580  POKE M,X:M = M + 1
590  GOTO 530
600  POKE M,A:M = M + 1
610  POKE M,0
620  RETURN
1000  DATA  2,2,2,7,4,4,7,7,4,4,5
     ,5,4,4,5,5,6,6,5,5,6,6,7,7,6
     ,6,7,8
```



รูปที่ 5.11

5.5 SUBROUTINE ในรูปที่ 5.4 นั้น สามารถจะเก็บ SHAPE TABLE ไว้ได้เพียงรูปเดียว โปรแกรมในรูปที่ 5.12 ข้างล่างนี้มี SUBROUTINE ที่สามารถเก็บรูปร่างไว้ได้หลายรูป

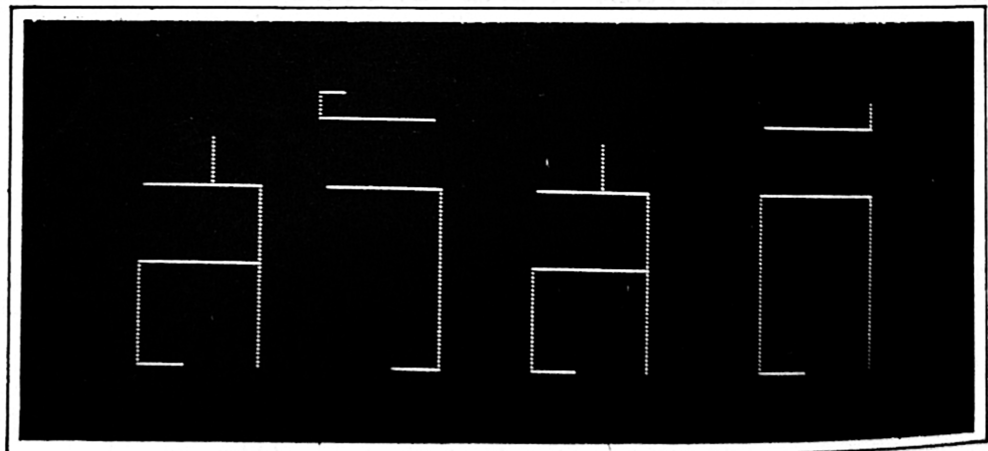
โปรแกรมนี้จะมีอยู่ 2 ส่วน บรรทัดที่ 10 ถึง 130 เป็น MAIN PROGRAM ใช้ในการสั่งให้เขียนรูปโดยกำหนด SCALE และการหมุน (ROT) ส่วนหลังตั้งแต่บรรทัดที่ 500 ไป เป็น SUBROUTINE
บรรทัดที่ 745 จะกำหนดจำนวนรูปร่าง ในที่นี้มี 5 รูป
บรรทัดที่ 750 ถึง 790 แต่ละบรรทัดคือ รูปร่างตามที่อธิบายไว้ในข้อ 5.2 แต่ละรูปร่างลงท้ายด้วย 8

```
10  REM   PRO#5.4
20  REM *TO PRINT SAWASDI*
30  HOME : HGR : HCOLOR= 3
34  VTAB 21
35  INPUT "ENTER SCALE < NOT MORE
      THAN 9  >  ";SC
40  GOSUB 500
50  SCALE= SC: ROT= 0
60  DRAW 1 AT 80,80
70  DRAW 2 AT 120,80
80  DRAW 1 AT 180,80
90  DRAW 3 AT 240,80
100  DRAW 4 AT 125,40
120  DRAW 5 AT 240,40
130  FOR I = 1 TO 5000: NEXT : TEXT
      : HOME : END
500  REM  PRO#5.4-SHAPE TABLES
505  REM  FOR MORE THAN 1 SHAPE
510 S = 768
520 S1 =  INT (S / 256):S2 = S -
     S1 * 256
530  POKE 232,S2: POKE 233,S1
540  READ N:K = N
550  POKE S,N: POKE S + 1,00
560 M = S + 2 * (N + 1):S2 = S +
     2
570 D = M - S
580  IF D > 255 THEN 600
585  POKE S2,D: POKE S2 + 1,0
590  GOTO 620
600 D1 =  INT (D / 256):D2 = D -
     D1 * 256
610  POKE S2,D2: POKE S2 + 1,D1
620 S2 = S2 + 2
630  READ A
640  IF A = 8 THEN 710
650  READ B
660  IF B = 8 THEN 700
670 X = B * 8 + A
680  POKE M,X:M = M + 1
690  GOTO 630
700  POKE M,A:M = M + 1
710  POKE M,0:M = M + 1
720 K = K - 1
730  IF K = 0 THEN  RETURN
```

```
740  GOTO 570
745  DATA  5
750  DATA   7,7,7,6,6,6,6,5,5,1,1
     ,1,4,4,4,4,7,7,5,5,4,4,4,7,7
     ,7,7,7,1,1,1,4,4,8
760  DATA   1,2,2,2,2,5,5,4,4,4,4
     ,4,4,4,7,7,7,7,7,8
770  DATA  3,2,2,2,2,7,7,4,4,4,4,
     4,4,4,5,5,5,5,5,6,6,6,6,6,6,
     6,8
780  DATA  3,3,7,6,5,5,5,5,5,8
790  DATA   1,1,6,7,7,7,7,7,8
```

รูปที่ 5.12

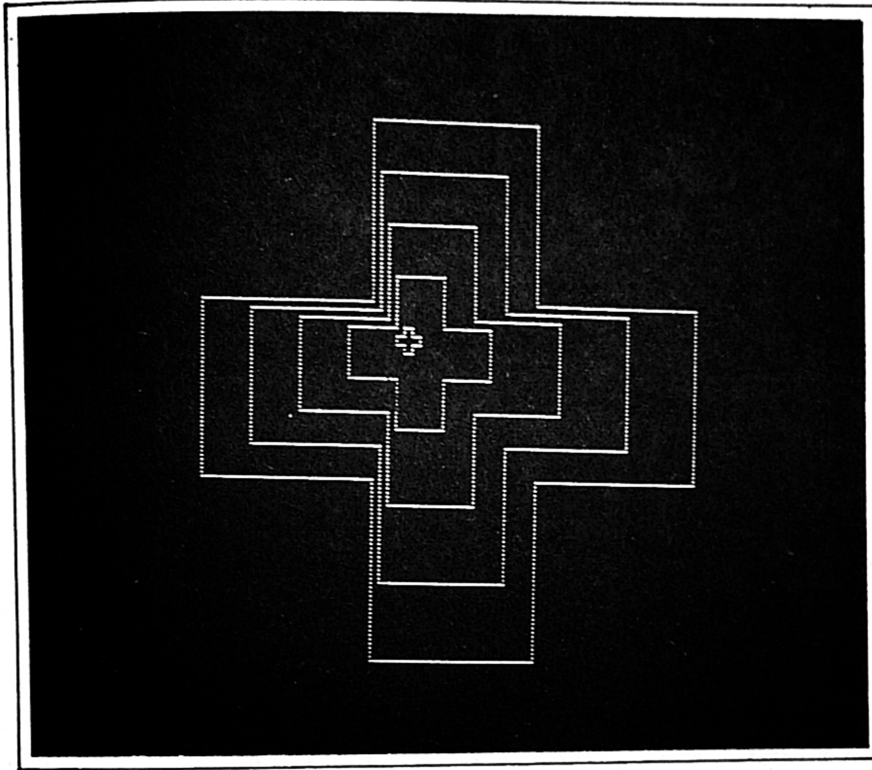
รูปที่ 5.13 คือผลที่ออกมาจากโปรแกรมดังกล่าว โดยกำหนดให้ SCALE เท่ากับ 8



รูปที่ 5.13

แบบฝึกหัดบทที่ 5

5.1 จงเขียนโปรแกรมเพื่อให้ปรากฏภาพบนจอดังนี้



5.2 จากโปรแกรมของแบบฝึกหัดข้อ 5.1 จงแก้โปรแกรมให้รูปขยายออกแล้วหมุนตัว

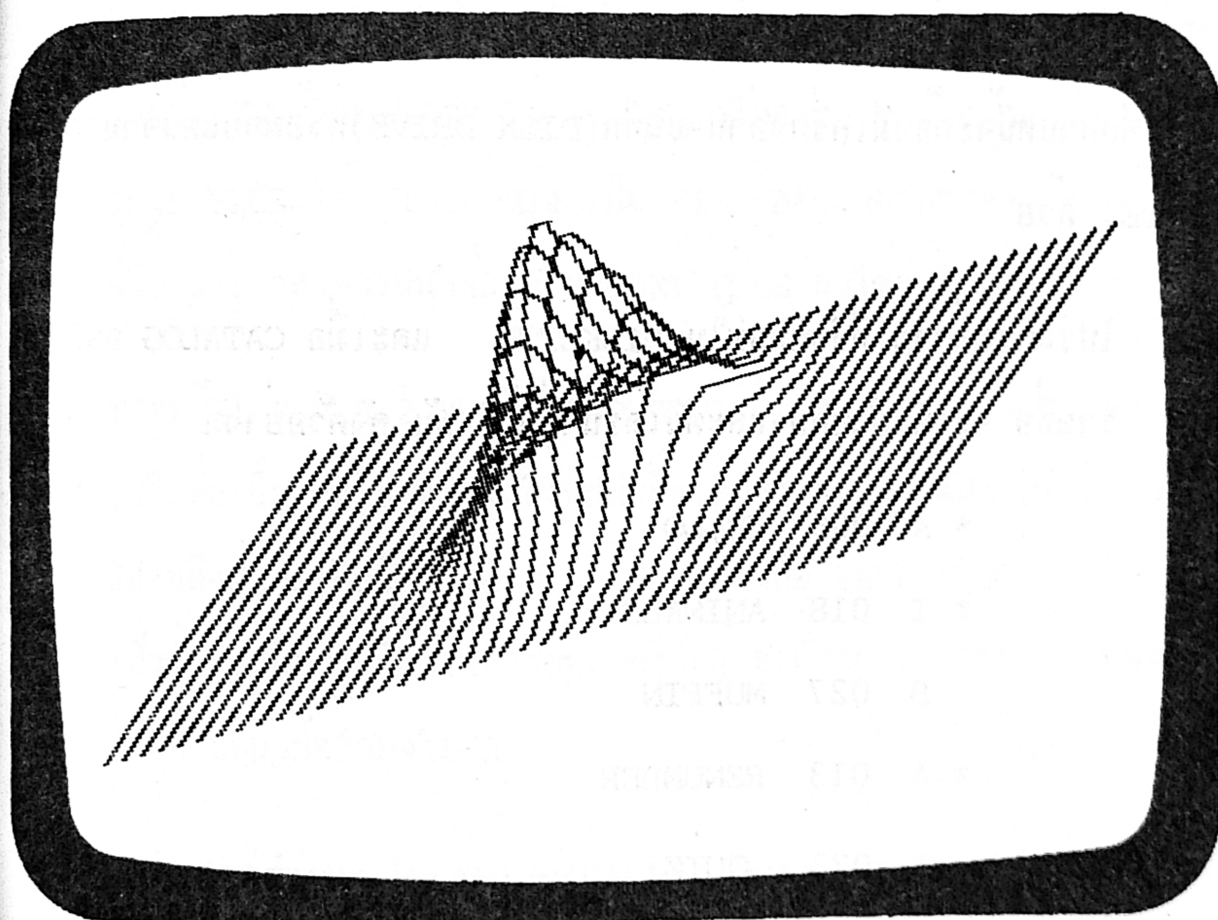
เฉลยแบบฝึกหัดบทที่ 5

5.1

```
10  REM  SOL#5.1
20  GOSUB 500
30  HOME : VTAB 22
40  HGR : HCOLOR= 3
50  ROT= 0:X = 140:Y = 80
60  FOR S = 1 TO 25 STEP 5
70  SCALE= S
80  DRAW 1 AT X,Y
90 X = X + 3:Y = Y + 3
100  NEXT
110  FOR I = 1 TO 3000: NEXT : TEXT
     : HOME : END
500  REM  PRO#5.1 SUBROUTINE FOR
     SHAPE TABLE
510  POKE 768,01: POKE 769,00: POKE
     770,04: POKE 771,00:M = 772
520  POKE 232,00: POKE 233,03
530  READ A
540  IF A = 8 THEN 610
550  READ B
560  IF B = 8 THEN 600
570 X = B * 8 + A
580  POKE M,X:M = M + 1
590  GOTO 530
600  POKE M,A:M = M + 1
610  POKE M,0
620  RETURN
1000  DATA  2,2,2,7,4,4,7,7,4,4,5
     ,5,4,4,5,5,6,6,5,5,6,6,7,7,6
     ,6,7,8
```

```
10  REM  SOL#5.1
20  GOSUB 500
30  HOME : VTAB 22
40  HGR : HCOLOR= 3
50 X = 140:Y = 80
60  FOR S = 1 TO 21 STEP 3
70  SCALE= S
75  ROT= S
80  DRAW 1 AT X,Y
82  GOSUB 200
85  XDRAW 1 AT X,Y
90 X = X + 3:Y = Y + 3
100  NEXT
110  FOR I = 1 TO 3: GOSUB 200: NEXT
     : TEXT : HOME : END
200  FOR I = 1 TO 1000: NEXT I: RETURN

500  REM  PRO#5.1 SUBROUTINE FOR
     SHAPE TABLE
510  POKE 768,01: POKE 769,00: POKE
     770,04: POKE 771,00:M = 772
520  POKE 232,00: POKE 233,03
530  READ A
540  IF A = 8 THEN 610
550  READ B
560  IF B = 8 THEN 600
570 X = B * 8 + A
580  POKE M,X:M = M + 1
590  GOTO 530
600  POKE M,A:M = M + 1
610  POKE M,0
620  RETURN
1000  DATA  2,2,2,7,4,4,7,7,4,4,5
     ,5,4,4,5,5,6,6,5,5,6,6,7,7,6
     ,6,7,8
```



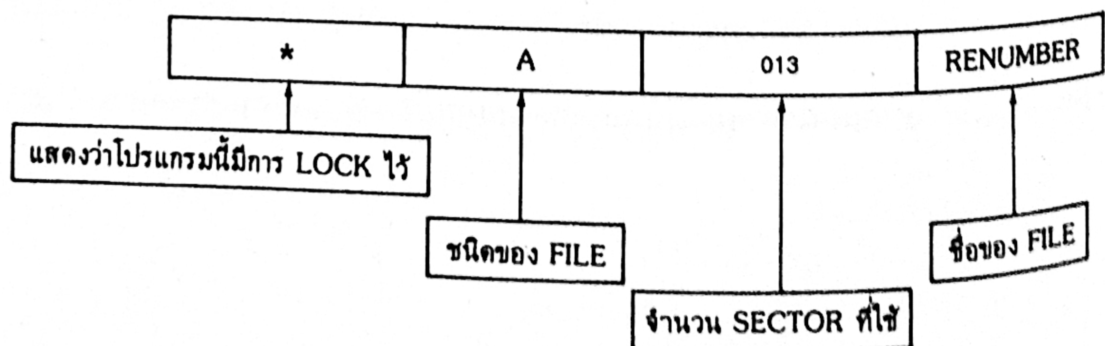
6 TEXT FILE

ผู้ที่จะศึกษาบทนี้จะต้องมีเครื่องอ่าน-บันทึก(DISK DRIVE)พร้อมกับแผ่นจาน(DISKETTE) ด้วย

6.1 โปรแกรมต่างๆ ที่บันทึกอยู่ใน DISKETTE และเมื่อ CATALOG ออกมาดูรายชื่อ จะเห็นตัวอักษรและตัวเลขนำต่างๆ กัน ดังตัวอย่างนี้

```
* A 006 HELLO
* I 018 ANIMALS
  B 027 MUFFIN
* A 013 RENUMBER
  T 023 QUIZ
```

ซึ่งจะมีความหมายดังนี้



รูปที่ 6.1

เรามาดูเฉพาะชนิดของ FILE ซึ่งมีอยู่ 4 จำพวกด้วยกันคือ

A คือ FILE ที่เขียนด้วยภาษา APPLESOFT BASIC

B คือ BINARY FILE เขียนด้วย MACHINE LANGUAGE

I คือ FILE ที่เขียนด้วยภาษา INTEGER BASIC

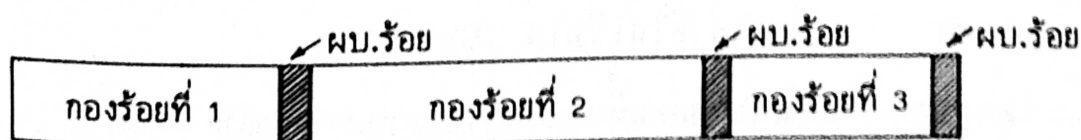
และ T คือ TEXT FILE หรือ DATA FILE

FILE A, B และ I นั้น เราสามารถใช้ SYSTEM COMMAND ให้ SAVE, LOAD, RUN และ LIST ได้ แต่เราไม่สามารถจะใช้คำสั่งดังกล่าวแก่ TEXT FILE ได้ TEXT FILE เป็น FILE อิสระแตกต่างจาก FILE ชนิดอื่น แต่เราสามารถนำข้อมูลใน TEXT FILE มาใช้กับโปรแกรมอื่นๆ ได้

TEXT FILE มีอยู่ 2 แบบคือ SEQUENTIAL TEXT FILE คือ FILE ที่บันทึกต่อเนื่องกันไปเหมือนกับเทปบันทึกเสียงซึ่งตั้งต้นจากปลายเทปข้างหนึ่งไปจนถึงปลายเทปอีกข้างหนึ่ง อีกแบบหนึ่งคือ RANDOM ACCESS FILE เป็นการบันทึกแบบที่ทำให้การค้นหาหรือแก้ไขเป็นไปอย่างสะดวก แต่จะกินเนื้อที่ในหน่วยความจำมาก

ความแตกต่างระหว่าง SEQUENTIAL FILE กับ RANDOM ACCESS FILE

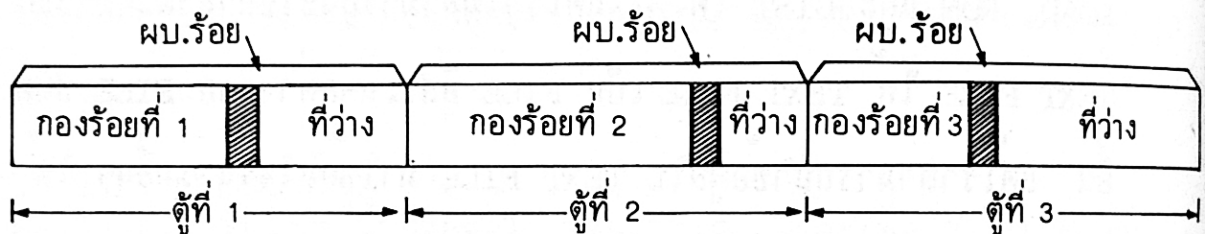
ทหารหนึ่งกองพันกำลังเดินเรียงสี่ไปสถานีรถไฟ ทหารกองพันนี้ประกอบไปด้วย 3 กองร้อย แต่ละกองร้อยมีจำนวนคนไม่เท่ากันในแถวนั้น แต่ละกองร้อยมีผู้บังคับกองร้อยเดินปิดท้าย แล้วต่อด้วยกองร้อยถัดไป ดังแสดงในรูป



รูปที่ 6.2

แถวทหารดังกล่าวนี้เสมือน FILE ที่เขียนแบบ SEQUENTIAL TEXT FILE กองร้อยเปรียบเสมือนกลุ่มตัวอักษร ผบ.ร้อยเปรียบเสมือนการกด RETURN, FILE แบบนี้จะต่อเนื่องกันไป จึงยากแก่การค้นหาหรือตรวจแก้

ทหารกองพันนี้เดินไปขึ้นรถไฟซึ่งทางรถไฟจัดตู้ให้ 3 ตู้แต่ละตู้สามารถบรรจุคนได้เท่าๆ กัน และมีที่นั่งมากกว่าจำนวนทหารแต่ละกองร้อย ทหารขึ้นไปนั่งบนรถกองร้อยแต่ละตู้ จึงมีที่ว่างเหลือในแต่ละตู้ดังในรูป



การตามหาพลทหารแดง ซึ่งอยู่กองร้อยที่ 3 ระหว่างการเดินแถวมาที่สถานี ย่อมจะยากกว่าเมื่อทหารขึ้นอยู่บนรถไฟแล้ว เช่นเดียวกับการหา(SEARCH) ข้อมูลใน SEQUENTIAL FILE ย่อมยากกว่าหาข้อมูลใน RANDOM ACCESS FILE แต่ข้อดีของ SEQUENTIAL FILE อยู่ที่ใช้เนื้อที่หน่วยความจำน้อยกว่า

6.2 **SEQUENTIAL TEXT FILE** เหมาะที่จะใช้บันทึกข่าวสารหรือข้อมูลที่ดำเนินต่อเนื่องตั้งแต่ต้นจนจบและไม่มีการแก้ไขมากนัก คำสั่งที่ใช้ในการเขียน อ่าน และบันทึกมีดังนี้

OPEN เป็นคำสั่งที่ให้เริ่มใช้ TEXT FILE

WRITE เป็นคำสั่งก่อนที่จะทำการป้อนข้อมูลเข้าใน FILE

PRINT เป็นคำสั่งที่ให้ถ่ายทอดข้อมูลจาก CPU เข้าในแผ่นจานบันทึก

READ เป็นคำสั่งที่ให้ถ่ายทอดข้อมูลจากแผ่นจานบันทึกเข้า CPU

CLOSE เป็นคำสั่งที่บอกว่าจะเลิกใช้ TEXT FILE แล้ว

คำทั้งหมดดังกล่าวนี้จะต้องตามด้วยชื่อของ FILE และการสั่งจะต้องสั่งด้วยคำสั่ง PRINT CHR$(4) ดังนี้ สมมติว่า FILE นี้ชื่อ FRUITS

```
50 PRINT CHR$(4) ; "OPEN FRUITS"
```

การสร้าง SEQUENTIAL FILE จะต้องเริ่มด้วยคำสั่ง OPEN ก่อน แล้วจบลงด้วย CLOSE

6.2.1 โปรแกรมในรูปที่ 6.3 เป็นโปรแกรมสร้าง FILE ชื่อ FRUITS

```
10  REM  PRO# 6.1
20  PRINT  CHR$ (4);"OPEN FRUITS"

30  PRINT  CHR$ (4);"WRITE FRUITS"

40  PRINT "APPLE"
50  PRINT "MANGO"
60  PRINT "ORANGE"
70  PRINT "BANANA"
80  PRINT "LONGAN"
90  PRINT  CHR$ (4);"CLOSE FRUITS"

100  END
```

โปรแกรมชื่อ FRUITS.

หลังจากพิมพ์โปรแกรมนี้เสร็จแล้ว ให้พิมพ์ RUN ข้อความทั้งหลายที่เขียนไว้ระหว่าง WRITE กับ CLOSE จะถูกบันทึกไว้ในแผ่นจาน

พิมพ์ CATALOG ออกมาดูจะพบว่าจะมี FILE ชื่อ FRUITS มีคำนำว่า T ปรากฏเพิ่มขึ้นมาในรายการโปรแกรม

มาพิจารณาโปรแกรมนี้จะเห็นว่าการ OPEN, WRITE หรือ CLOSE จะต้องสั่งผ่านคำสั่ง PRINT CHR$(4)ซึ่งเป็น SYSTEM COMMAND ที่จะต้องบรรจุไว้ในโปรแกรม(ดูข้อ 2.6) การที่จะเขียนคำสั่งนี้บ่อยๆ เป็นเรื่องน่ารำคาญ เราอาจจะย่อโดยกำหนดให้ D$ = CHR$(4) เราจะเขียนได้ง่ายกว่า

ในบางครั้ง FILE ชื่อ FRUITS อาจจะมีอยู่ก่อนแล้ว หากเขียนทับลงไปข้อความใน FILE เก่าจะเข้ามาปนกับของใหม่ ดังนั้นการเขียนจึงมักจะลบ FILE เดิมออกแล้วเปิด FILE ใหม่ในรายชื่อของ FILE เดิม โปรแกรมในรูปที่ 6.4 เป็นโปรแกรมที่แก้ไขใหม่

```
10  REM  PRO#6.2
20 D$ =  CHR$ (4)
30  PRINT D$;"OPEN FRUITS"
40  PRINT D$;"DELETE FRUITS"
50  PRINT D$;"OPEN FRUITS"
60  PRINT D$;"WRITE FRUITS"
70  PRINT "APPLE"
80  PRINT "ORANGE"
90  PRINT D$;"CLOSE FRUITS"
100  END
```

โปรแกรมชื่อ FRUITS แก้ไขใหม่

6.2.2 โปรแกรมในรูปที่ 6.5 แสดงการสร้างโปรแกรมเพื่อเขียน TEXT FILE

```
10  REM  PRO#6.3
20  REM  * HOW TO WRITE A SEQUENT
     IAL FILE *
25  HOME
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN INVENTORY"
50  PRINT D$;"DELETE INVENTORY"
60  PRINT D$;"OPEN INVENTORY"
70  PRINT
80  INPUT "ENTER ITEM DESCRIPTION
     (TYPE 'STOP' IF NO MORE DATA
     ): ";DE$
90  IF DE$ = "STOP" THEN 160
100  INPUT "QUANTITY OF ITEM :";N

110  INPUT "VALUE OF EACH :";V
120  PRINT D$;"WRITE INVENTORY"
130  PRINT DE$: PRINT N: PRINT V
140  PRINT D$
150  GOTO 70
160  PRINT D$;"CLOSE INVENTORY"
170  END
```

โปรแกรมสร้าง FILE ชื่อ "WRITE FILE"

SAVE โปรแกรมนี้ไว้ในชื่อ "WRITE FILE" โปรแกรมจะเข้าไปบันทึกอยู่ในแผ่นจานในรูปของ PROGRAM FILE ซึ่งต่อไปเราจะนำโปรแกรมนี้ไปเขียน TEXT FILE ซึ่งจะมีชื่อว่า "INVENTORY"

ก่อนที่เราจะนำไปใช้ เรามาพิจารณาโปรแกรมนี้กันก่อน

บรรทัดที่ 30 ถึง 60 เป็นพิธีการเปิด TEXT FILE โดยให้ชื่อว่า INVEN-TORY

บรรทัดที่ 80 ถึง 110 เป็นการป้อนข้อมูลช่วงแรกเข้าไปในเครื่อง

บรรทัดที่ 120 เป็นคำสั่งที่ให้เขียน FILE

บรรทัดที่ 130 เป็นการเขียน FILE โดยนำข้อมูลจากบรรทัดที่ 80 ถึง 110 ที่ป้อนเข้าไป เอาเข้าไปบันทึกไว้ในแผ่นจาน

บรรทัดที่ 140 เป็นคำสั่งที่หยุดการเขียน FILE โปรดสังเกตคำสั่งนี้ซึ่งจะนำไปใช้ในโอกาสต่อไป

บรรทัดที่ 150 ให้ย้อนกลับไปบรรทัดที่ 70 เริ่มป้อนข้อมูลช่วงที่ 2 เข้าไปแล้วดำเนินต่อไปทีละช่วง โดยโปรแกรมจะหมุนเป็นวงจรจนกระทั่งหมดข้อมูล ผู้ใช้จะพิมพ์คำว่า STOP ลงไปซึ่งจะทำให้โปรแกรมกระโดดไปที่บรรทัดที่ 160 ซึ่งเป็นคำสั่งที่ให้ปิด FILE แล้วจบโปรแกรมในบรรทัดที่ 170

เมื่อตรวจโปรแกรมนี้และ SAVE ไว้เข้าบัญชีเป็น PROGRAM FILE แล้วให้พิมพ์ RUN ต่อจากนี้ไปจะเป็นการสร้าง TEXT FILE ในชื่อ INVENTORY เครื่องจะให้ป้อนข้อมูลเข้าไป สมมติว่าข้อมูลชุดแรกเป็นดังนี้

ITEM DESCRIPTION	TABLES
QUANTITY OF ITEM	10
VALUE OF EACH	500

ชุดต่อๆ ไปมีดังนี้

CHAIRS	25	200
ELEC. FAN	5	1200
BOOK SHELVES	3	800

และเมื่อหมดข้อมูลแล้วให้พิมพ์ STOP DISK DRIVE จะหมุน แสดงว่ารับข้อมูลทั้งหมด แล้วนำไปเขียนลงไว้ในแผ่นจาน

เมื่อ DRIVE หยุดหมุนให้พิมพ์ CATALOG ตรวจดูจะพบว่ามี FILE ปรากฏต่อท้ายรายการดังนี้

T 002 INVENTORY

FILE ที่ปรากฏใหม่นี้คือ TEXT FILE

ลองพิมพ์ LOAD INVENTORY จะพบว่าเครื่องจะบอกว่า FILE TYPE MISMATCH ลอง RUN INVENTORY ก็เหมือนเดิม แสดงว่า TEXT FILE นี้ LOAD ไม่ได้ RUN ไม่ได้ แต่สามารถจะ LOCK และ DELETE ได้

แล้วเราจะเอา FILE นี้มาใช้ได้อย่างไร

โปรแกรมในรูปที่ 6.6 จะเป็นการอ่านข้อมูลใน TEXT FILE ชื่อ INVENTORY แล้วนำเอามาพิมพ์ให้ดูบนจอ

```
10  REM  PRO#6.4
20  HOME : REM  * HOW TO READ AND
      PRINT FILE *
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN INVENTORY"
50  PRINT "DESCRIPTION"; TAB( 18)
     ;"QUANTITY"; TAB( 30);"VALUE
      -BAHT"
60  PRINT
65  ONERR  GOTO 120
70  PRINT D$;"READ INVENTORY"
80  INPUT DE$,N,V
90  PRINT D$
100  PRINT DE$; TAB( 22);N; TAB(
     33);V
110  GOTO 70
120  IF  PEEK (222) = 5 THEN  PRINT
     D$;"CLOSE INVENTORY": END
```

โปรแกรมอ่าน SEQUENTIAL FILE ชื่อ "READ AND PRINT FILE"

เรามาสังเกตโปรแกรมนี้ตามลำดับ

บรรทัดที่ 40 เป็นการ OPEN FILE ตอนนี้ต้องระวังอย่าเผลอไปเพิ่มบรร-ทัดที่ DELETE FILE ชื่อ INVENTORY เข้า FILE ที่เราบรรจงเขียนจะหายไปหมด

บรรทัดที่ 50 เป็นคำสั่ง PRINT ธรรมดา เพื่อพิมพ์หัวข้อ

บรรทัดที่ 65 เป็นบรรทัดที่ต้องเขียนไว้ในโปรแกรมนี้ มิฉะนั้นเมื่อเครื่องอ่านจนถึงข้อมูลสุดท้ายแล้วเครื่องจะบอก ERROR (ดูบทที่ 2 ข้อ 2.7)

บรรทัดที่ 70 เป็นการอ่านข้อมูลจาก TEXT FILE ชื่อ INVENTORY ซึ่งเราได้บรรจุไว้แล้ว

บรรทัดที่ 80 เป็นการนำข้อมูลใน TEXT FILE ชื่อ INVENTORY ซึ่งอยู่ในแผ่นจาน(DISKETTE) ชุดแรกถ่ายเข้ามาไว้ใน CPU เมื่อหมดข้อมูลชุดแรก

บรรทัดที่ 90 จะสั่งหยุดการอ่าน

บรรทัดที่ 100 จะเอาข้อมูลชุดแรกใน CPU มาพิมพ์ลงบนจอภาพ

บรรทัดที่ 110 เป็นคำสั่งที่สั่งให้ไปเริ่มอ่านข้อมูลชุดที่ 2 แล้วนำมาพิมพ์อีกหมุนไปจนหมดข้อมูล ซึ่งหลังจากหมดข้อมูลแล้วเครื่องจะบอกว่า OUT OF DATA ERROR แล้วหยุดทำงาน แต่บรรทัดที่ 65 และ 120 จะช่วยไม่ให้เกิดเรื่องดังนี้ขึ้น

บรรทัดที่ 120 จะยกเลิกคำสั่งในบรรทัดที่ 65 คือหากมี ERROR อื่นให้บอกแต่ถ้าเป็น ERROR ที่เรียกว่า "OUT OF DATA ERROR" ให้ปิด FILE แล้วจบโปรแกรม (ดูเพิ่มเติมในข้อ 6.2.3)

SAVE โปรแกรมนี้ไว้ในชื่อโปรแกรมที่ไม่ใช่ INVENTORY (เพราะชื่อนี้สงวนไว้ให้เป็นชื่อของ TEXT FILE แล้ว) สมมติว่าโปรแกรมนี้ชื่อ "READ AND PRINT FILE"

พิมพ์ CATALOG ออกมาดู จะเห็นโปรแกรมสุดท้ายเรียงกันอยู่ 3 โปรแกรม คือ

A 003 WRITE FILE

T 002 INVENTORY

A 003 READ AND PRINT FILE

โปรแกรมแรกที่ชื่อ "WRITE FILE" เป็น PROGRAM FILE ที่เขียนไว้สำหรับการสร้าง TEXT FILE ที่ชื่อ INVENTORY ขึ้น ส่วนโปรแกรมหลังที่ชื่อ READ AND PRINT FILE เป็น PROGRAM FILE เช่นกัน แต่เขียนไว้สำหรับเอา TEXT FILE ที่ชื่อ INVENTORY มาใช้คือ นำมาแสดงบนจอภาพตามรูปแบบที่เรากำหนดให้

6.2.3 ลองเขียนโปรแกรมเต็มรูปแบบเพื่อเก็บชื่อและหมายเลขโทรศัพท์

```
10  REM  PRO#6.5
20  HOME : REM   * PHONE *
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN PHONE LIST"
50  PRINT D$;"DELETE PHONE LIST"
60  PRINT D$;"OPEN PHONE LIST"
65  PRINT
70  PRINT "TYPE 'STOP' IF NO MORE
      DATA": PRINT
80  INPUT "ENTER NAME (NOT MORE T
     HAN 25 CHARACTERS):";N$
85  IF N$ = "STOP" THEN 290
```

```
90  IF  LEN (N$) > 25 THEN  PRINT
     CHR$ (7);"NOT MORE THAN 25
    CHAR. PLEASE": PRINT : GOTO
    80
100  IF  LEN (N$) < 25 THEN N$ =
    N$ + " ": GOTO 100
110  PRINT "PHONE NUMBER FORMAT:
    XXX-XXXX"
120  INPUT "ENTER THE NUMBER: ";P
    $
130  IF  LEN (P$) <  > 8 THEN  PRINT
     CHR$ (7);"FORMAT MISTAKE-PL
    EASE REENTER": PRINT : GOTO
    120
140  IF  ASC ( MID$ (P$,4,1)) <  >
    45 THEN  PRINT  CHR$ (7);"FO
    RMAT MISTAKE,USE '-' IN BETW
    EEN ": GOTO 120
150  HOME : FLASH : PRINT "PLEASE
     CHECK YOU ENTRY": NORMAL
160  PRINT "NAME: ";N$
170  PRINT "PHONE NO.: ";P$
180  INPUT "ARE THEY CORRECT ? ";
    R$
190  IF R$ <  > "Y" AND R$ <  > "
    N" THEN  PRINT  CHR$ (7);"PL
    EASE ENTER 'Y' OR 'N' ": PRINT
    : GOTO 180
200  IF R$ = "Y" THEN  HOME : GOTO
    240
210  IF R$ = "N" THEN  PRINT : PRINT
    "PLEASE REENTER NAME AND PHO
    NE NO."
230  GOTO 65
240 M$ = N$ + P$
250  PRINT D$;"WRITE PHONE LIST"
260  PRINT M$
270  PRINT D$
280  GOTO 70
290  PRINT D$;"CLOSE PHONE LIST"
300  PRINT  TAB( 12)"END OF LISTI
    NG"
310  END
```

รูปที่ 6.7 โปรแกรมสร้าง SEQUENTIAL FILE ชื่อ "PHONE"

พิมพ์แล้ว SAVE โปรแกรมนี้ในชื่อของ "PHONE" (โปรดสังเกตว่าภายในโปรแกรมนี้มีสิ่งที่น่าสนใจหลายอย่าง เช่น เครื่องจะบังคับให้พิมพ์ชื่อได้ไม่เกิน 25 ตัวอักษร รูปแบบของโทรศัพท์ต้องเป็นไปตามที่กำหนดคือ ตัวเลขกลุ่มหน้ามี 3 ตัว มีขีดคั่นแล้วตัวเลขกลุ่มหลังมี 4 ตัว เครื่องจะบังคับให้เป็นไปตามนี้ การถามจะต้องตอบให้ตรง ฯลฯ ผู้เรียนพึงสังเกตแล้วนำไปใช้)

เมื่อเขียนโปรแกรมนี้เสร็จให้พิมพ์ RUN เครื่องจะให้ป้อนข้อมูลคือ ชื่อและหมายเลขโทรศัพท์ เมื่อหมดให้พิมพ์ STOP DISK DRIVE จะหมุน แสดงว่าเครื่องรับข้อมูลทั้งหมดเข้าไปบันทึกไว้ในแผ่นจานสร้าง TEXT FILE ในชื่อของ PHONE LIST เมื่อจบลอง CATALOG ดูจะเห็น TEXT FILE นี้ปรากฏอยู่ในท้าย CATALOG ขณะนี้จะมีอะไรไปอยู่ใน FILE บ้าง เรายังเรียกออกมาดูไม่ได้ จึงต้องมีการเขียนโปรแกรมเอา TEXT FILE นี้มาดูโปรแกรมในรูปที่ 6.8

```
10  REM  PRO#6.6
20  HOME : PRINT  TAB( 12);"PHONE
      DIRECTORY": PRINT
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN PHONE LIST"
50  ONERR  GOTO 110
60  PRINT D$;"READ PHONE LIST"
70  INPUT M$
80  PRINT D$
90  PRINT M$: PRINT
100  GOTO 60
110  IF  PEEK (222) = 5 THEN 130
120  POKE 216,0
130  PRINT D$;"CLOSE PHONE LIST"
140  PRINT  TAB( 12)" END OF FILE
     "
150  END
```

รูปที่ 6.8 โปรแกรมอ่าน SEQUENTIAL FILE ชื่อ "PHONE DIRECTORY"

โปรแกรมนี้มีคำสั่งที่น่าสนใจคือบรรทัดที่ 120 คำสั่งนี้จะยกเลิกคำสั่ง บรรทัดที่ 50 หมายความว่าตั้งแต่บรรทัดนี้(120) ไปแล้ว หากมี ERROR อื่นใดที่ไม่ใช่ ERROR หมายเลข 5 (คือ END OF DATA ERROR)แล้วให้บอกด้วยว่าเป็น ERROR ชนิดใด แต่ถ้าเป็น ERROR หมายเลข 5 แล้ว บรรทัดที่ 110 จะสั่งให้ไปที่บรรทัดที่ 130 คือปิด FILE (ดูรายละเอียดในหนังสือ THE DOS MANUAL หน้า 114, 115)

เมื่อพิมพ์โปรแกรมนี้แล้วให้ SAVE ไว้ในชื่อของ "PHONE DIRECTORY"

ทดลองใช้โปรแกรมในรูปที่ 6.7 เขียน FILE โดยปฏิบัติดังนี้

1. LOAD โปรแกรมที่ชื่อ PHONE (โปรแกรมในรูปที่ 6.7)
2. RUN โปรแกรมนี้ แล้วป้อนข้อมูลเหล่านี้ลงไป

```
SE. EDUCATION              245-0309
K. ENG. CONSULTANTS        277-0149
MIN. OF FINANCE            221-0889
MIN. OF COMMUNICATION      281-3422
STOP
```

3. พิมพ์ CATALOG จะเห็น TEXT FILE ซึ่ง PHONE LIST
4. พิมพ์โปรแกรมในรูปที่ 6.8 แล้ว RUN ดู จะปรากฏชื่อ FILE ที่พิมพ์ไว้ในข้อ 2 ดังนี้

```
            PHONE DIRECTORY

SE.EDUCATION                245-0390

K.ENG.CONSULTANTS           277-0149
```

```
MIN.OF FINANCE                  221-0889

MIN.OF COMMUNICATION            281-3422

              END OF FILE
```

6.2.4 โปรแกรมในรูปที่ 6.7 นั้นเมื่อได้ป้อนข้อมูลลงไปและปิด FILE แล้ว หากมีการเพิ่มข้อมูลอีกจะทำไม่ได้ เพราะถ้า RUN โปรแกรมนี้อีกครั้ง ข้อมูลซึ่งอยู่ใน TEXT FILE เดิมจะถูกลบ (ดูบรรทัดที่ 50 ของโปรแกรมในรูปที่ 6.7)

วิธีที่จะเพิ่มข้อมูลเพิ่มเติมใน TEXT FILE ทำได้โดยใช้คำสั่ง APPEND และโดยการเอาโปรแกรมในรูปที่ 6.7 มาแก้ และ SAVE ไว้ในชื่อใหม่ดังนี้

1. LOAD โปรแกรมในรูปที่ 6.7 ในชื่อ "PHONE" LIST ออกมาดูบรรทัดที่ 40 ถึง 60 ในโปรแกรมนี้มีดังนี้

```
40   PRINT D$;"OPEN PHONE LIST"
50   PRINT D$;"DELETE PHONE LIST"
60   PRINT D$;"OPEN PHONE LIST"
```

2. แก้บรรทัดที่ 50 และ 60 ดังนี้

```
50   PRINT D$;"CLOSE PHONE LIST"
60   PRINT D$;"APPEND PHONE LIST"
```

หลังจากแก้แล้วให้ SAVE ไว้ในชื่อของ "PHONE-APPEND" ซึ่งจะเป็นดังนี้

```
10  REM   PRO#6.7
20  HOME : REM    * PHONE-APPEND
     *
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN PHONE LIST"
50  PRINT D$;"CLOSE PHONE LIST"
60  PRINT D$;"APPEND PHONE LIST"
65  PRINT
70  PRINT "TYPE 'STOP' IF NO MORE
      DATA": PRINT
80  INPUT "ENTER NAME (NOT MORE T
     HAN 25 CHARACTERS):";N$
85  IF N$ = "STOP" THEN 290
90  IF  LEN (N$) > 25 THEN  PRINT
      CHR$ (7);"NOT MORE THAN 25
     CHAR. PLEASE": PRINT : GOTO
     80
100  IF  LEN (N$) < 25 THEN N$ =
     N$ + " ": GOTO 100
110  PRINT "PHONE NUMBER FORMAT:
     XXX-XXXX"
120  INPUT "ENTER THE NUMBER: ";P
     $
130  IF  LEN (P$) <  > 8 THEN  PRINT
      CHR$ (7);"FORMAT MISTAKE-PL
     EASE REENTER": PRINT : GOTO
     120
140  IF  ASC ( MID$ (P$,4,1)) <  >
     45 THEN  PRINT  CHR$ (7);"FO
     RMAT MISTAKE,USE '-' IN BETW
     EEN ": GOTO 120
150  HOME : FLASH : PRINT "PLEASE
      CHECK YOU ENTRY": NORMAL
160  PRINT "NAME: ";N$
170  PRINT "PHONE NO.: ";P$
180  INPUT "ARE THEY CORRECT ? ";
     R$
190  IF R$ <  > "Y" AND R$ <  > "
     N" THEN  PRINT  CHR$ (7);"PL
     EASE ENTER 'Y' OR 'N' ": PRINT
     : GOTO 180
200  IF R$ = "Y" THEN  HOME : GOTO
     240
```

```
210  IF R$ = "N" THEN  PRINT : PRINT
     "PLEASE REENTER NAME AND PHO
     NE NO."
230  GOTO 65
240 M$ = N$ + P$
250  PRINT D$;"WRITE PHONE LIST"
260  PRINT M$
270  PRINT D$
280  GOTO 70
290  PRINT D$;"CLOSE PHONE LIST"
300  PRINT  TAB( 12)"END OF LISTI
     NG"
310  END
```

รูปที่ 6.9 โปรแกรมเพิ่มเติม SEQUENTIAL FILE ชื่อ "PHONE - APPEND"

ใช้โปรแกรมนี้โดยการ RUN แล้วเพิ่มเติมชื่อและหมายเลขโทรศัพท์ขึ้นอีก 2 รายการคือ

HABOUR DEPT.	233-1311
POLICE DEPT.	252-7909

แล้วพิมพ์ STOP เครื่องจะรับหมายเลขโทรศัพท์ใหม่นี้ต่อเข้ากับรายการเดิม ลองใช้โปรแกรมในรูปที่ 6.8 "PHONE DIRECTOR" RUN ออกมาจะพบว่าเป็นดังนี้

```
          PHONE DIRECTORY

SE.EDUCATION                 245-0390

K.ENG.CONSULTANTS            277-0149

MIN.OF FINANCE               221-0889

MIN.OF COMMUNICATION         281-3422
```

```
HARBOUR DEPT.                233-1311

POLICE DEPT.                 252-7909

              END OF FILE
```

รูปที่ 6.10 โปรแกรมชื่อ PHONE DIRECTORY สร้างจาก SEQUENTIAL FILE

6.3 **RANDOM ACCESS FILE** ย้อนกลับไปดู "ความแตกต่างระหว่าง SEQUENTIAL FILE กับ RANDOM FILE" ที่กล่าวไว้ใน 6.2 ตู้รถไฟตู้หนึ่งเรียกว่า RECORD การใช้คำสั่ง OPEN ใน RANDOM ACCESS FILE จะต้องกำหนดความยาวของ RECORD เป็นจำนวน BYTE และการใช้คำสั่ง READ และ WRITE ต้องกำหนดหมายเลขของ RECORD ไว้ด้วยดังนี้

```
50  PRINT CHR$(4) ; "OPEN FRUITS, L 200"
60  PRINT CHR$(4) ; "WRITE FRUITS, R 1"
```

บรรทัดที่ 50 ได้มีการกำหนดให้ FILE ที่มีชื่อว่า FRUITS นี้มีความยาวแต่ละ RECORD เท่ากับ 200 BYTES

บรรทัดที่ 60 จะมีการเขียน FILE ที่ชื่อ FRUITS ใน RECORD ที่ 1

6.3.1 เราจะนำโปรแกรมในรูปที่ 6.7 ซึ่งเขียนชื่อและหมายเลขโทรศัพท์มาเขียนด้วย RANDOM ACCESS FILE

ก่อนอื่นเราต้องกำหนดความยาวของ RECORD ก่อน สำหรับชื่อเรากำหนดไว้ไม่ให้เกินกว่า 25 ตัวอักษร หมายเลขโทรศัพท์มี 8 ตัว รวม 33 บวก "เผื่อขาด" อีกกลุ่มละ 1 เป็น 35 ตัวอักษร หรือ 35 BYTES

เพื่อป้องกันไม่ให้สับสนกับโปรแกรมเดิมที่เขียนแบบ SEQUENTIAL FILE เราจะกำหนดชื่อทั้งหมดให้เหมือนกับของเดิม แต่มีเลข 2 ต่อท้ายโปรแกรม ในรูปที่6.11 เรานำโปรแกรมในรูปที่ 6.7 ซึ่งเป็น SEQUENTIAL FILE มาดัดแปลงเป็น RANDOM ACCESS FILE ส่วนที่เพิ่มเติมหรือเปลี่ยนแปลง (นอกจากชื่อโปรแกรม) ได้ขีดเส้นใต้ไว้

```
10  REM  PRO#6.8
20  HOME : REM  *PHONE2*
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN PHONE LIST2"
50  PRINT D$;"DELETE PHONE LIST2"

60  PRINT D$;"OPEN PHONE LIST2,L3
    5"
70 R = 0
80  PRINT
90  PRINT "TYPE 'STOP' IF NO MORE
     DATA": PRINT
100  INPUT "ENTER NAME (NOT MORE
     THAN 25 CHARACTERS)";N$
110  IF N$ = "STOP" THEN 320
120 R = R + 1
130  IF  LEN (N$) > 25 THEN  PRINT
      CHR$ (7);"NOT MORE THAN 25
     CHAR. PLEASE": PRINT : GOTO
     100
140  IF  LEN (N$) < 25 THEN N$ =
     N$ + " ": GOTO 140
150  PRINT "PHONE NUMBER FORMAT:
     XXX-XXXX"
160  INPUT "ENTER THE NUMBER: ";P
     $
170  IF  LEN (P$) <  > 8 THEN  PRINT
      CHR$ (7);"FORMAT MISTAKE, P
     LEASE REENTER": PRINT : GOTO
     160
180  IF  ASC ( MID$ (P$,4,1)) <  >
     45 THEN  PRINT  CHR$ (7);"FO
     RMAT MISTAKE, USE'-' IN BETW
     EEN": GOTO 160
```

```
190  HOME : FLASH : PRINT "PLEASE
      CHECK YOUR ENTRY": NORMAL
200  PRINT "NAME: ";N$
210  PRINT "PHONE NO.:  ";P$
220  INPUT "ARE THEY CORRECT? ";R
     $
230  IF R$ <  > "Y" AND R$ <  > "
     N" THEN  PRINT  CHR$ (7);"PL
     EASE ENTER 'Y' OR 'N ' ": PRINT
     : GOTO 220
240  IF R$ = "Y" THEN  HOME : GOTO
     270
250  IF R$ = "N" THEN  PRINT : PRINT
     "PLEASE REENTER NAME AND PHO
     NE NO."
260  GOTO 80
270 M$ = N$ + P$
280  PRINT D$;"WRITE PHONE LIST2,
     R";R
290  PRINT M$
300  PRINT D$
310  GOTO 90
320  PRINT D$;"WRITE PHONE LIST2,
     R0"
330  PRINT R
340  PRINT D$
350  PRINT D$;"CLOSE PHONE LIST2"

360  PRINT : PRINT  TAB( 12)"END
     OF LISTING"
370  END
```

รูปที่ 6.11 โปรแกรมสร้าง RANDOM ACCESS FILE ชื่อ "PHONE 2"

บรรทัดที่ 60 เพิ่มคำว่า ,L35 ลงท้ายชื่อโปรแกรม เพื่อกำหนดให้แต่ละ RECORD มีที่ไม่เกิน 35 BYTE

บรรทัดที่ 70 ลบล้างให้หมายเลขของ R ให้เท่ากับ 0 เป็นการเริ่มต้น

บรรทัดที่ 120 กำหนดให้ชื่อแรก R จะเท่ากับ 1 หมายความว่าเมื่อถึงบรร-

ทัดที่ 280 ชื่อแรกจะถูกบันทึกไว้ใน RECORD ที่ 1 เมื่อโปรแกรมกลับมาบรรทัดที่ 90 R จะเป็น 2 ในบรรทัดที่ 120 ชื่อที่ 2 จะถูกบันทึกไว้ใน RECORD ที่ 2 ตามลำดับ

บรรทัดที่ 320-340 เป็นการบันทึกว่าได้มีการเขียนลงใน FILE ไปกี่ RECORD แล้ว เพื่อประโยชน์ในการอ่าน FILE ทีหลัง บันทึกดังกล่าวนี้จะเขียนไว้ใน RECORD หมายเลข 0

```
]LIST

10  REM   PRO#6.9
20  HOME : PRINT  TAB( 12);"PHONE
     DIRECTORY2": PRINT
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN PHONE LIST2,L3
    5"
50  PRINT D$;"READ PHONE LIST2,R0
    "
60  INPUT R
70  PRINT D$
80  FOR I = 1 TO R
90  PRINT D$;"READ PHONE LIST2,R"
    I
100  INPUT M$
110  PRINT D$
120  PRINT M$: PRINT
130  NEXT I
140  PRINT D$;"CLOSE"
150  PRINT  TAB( 12)"END OF FILE"

160  END
```

รูปที่ 6.12 โปรแกรมอ่าน RANDOM ACCESS FILE ชื่อ "PHONE DIRECTORY 2"

SAVE โปรแกรมนี้ในชื่อของ PHONE2 แล้ว RUN โปรแกรมพร้อมกับป้อนข้อมูลต่อไปนี้ลงไปเพื่อบันทึกไว้ใน RANDOM ACCESS TEXT FILE ที่มีชื่อว่า PHONE LIST2

KIMENG	SAEPHU	123-4567
PRATOOM	PIYAKIJ	567-8900
CHAO	SAEDAN	234-6020
STOP		

เสร็จแล้วลอง CATALOG ดู จะพบว่ามี TEXT FILE ในชื่อ "PHONE LIST2" ปรากฏอยู่

6.3.2 โปรแกรมในรูปที่ 6.12 เป็นโปรแกรมที่จะนำเอา TEXT FILE ที่ชื่อ "PHONE LIST2" มาพิมพ์เป็นรูปแบบ สิ่งที่น่าสนใจในโปรแกรมนี้มีดังนี้

บรรทัดที่ 40 บอกจำนวนความยาวของ RECORD ว่าไม่เกิน 35 BYTES

บรรทัดที่ 50 เป็นการอ่านบันทึกใน RECORD หมายเลข 0 ว่าได้มีการป้อนข้อมูลลงไปกี่ RECORD

บรรทัดที่ 80-130 เป็นการอ่านข้อมูลในแต่ RECORD เริ่มแต่ RECORD ที่ 1 จนถึง RECORD ที่ R

บรรทัดที่ 140 เป็นการพิมพ์ข้อมูลของแต่ละ RECORD

SAVE โปรแกรมนี้ไว้ในชื่อของ PHONE DIRECTORY2 แล้ว RUN โปรแกรมนี้จะเรียก TEXT FILE ที่ชื่อ "PHONE LIST2" มาพิมพ์ ดังแสดงในรูปที่ 6.13

```
]RUN
              PHONE DIRECTORY2

KIMENG SAEPHU                    123-4567

PRATOOM PIYAKIJ                  567-8900

CHAO SAEDAN                      234-6020

              END OF FILE
```

รูปที่ 6.13 โปรแกรม PHONE DIRECTORY 2 สร้างจาก RAMDOM ACCESS FILE

6.3.3 การเพิ่มเติมรายการเข้าไปอีกใน FILE ที่ปิดแล้ว ใน RANDOM ACCESS FILE ไม่ได้ใช้คำสั่ง APPEND แต่นำจำนวนหมายเลขของข้อมูลที่บันทึกไว้ใน RECORD หมายเลข 0 มาใช้ดังโปรแกรมที่แสดงในรูปที่ 6.14

```
]PR#0
]LIST

10  REM  PRO#6.10
20  HOME : REM  * ADD PHONE2 *
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN PHONE LIST2,L3
     5"
50  PRINT D$;"READ PHONE LIST2,RØ
     "
60  INPUT R
70  PRINT D$
80  PRINT : PRINT "RECORD NO: ";R
     : PRINT
90 R = R + 1
100  INPUT "ENTER NAME (NOT MORE
     THAN 25 CHARACTERS):";N$
110  IF  LEN (N$) > 25 THEN  PRINT
      CHR$ (7);"NOT MORE THAN 25
     CHAR.,PLEASE": GOTO 100
```

```
120  IF  LEN (N$) < 25 THEN N$ =
     N$ + " ": GOTO 120
130  PRINT "PHONE NUMBER FORMAT:
     XXX-XXXX"
140  INPUT "ENTER THE NUMBER:";P$

150  IF  LEN (P$) <  > 8 THEN  PRINT
      CHR$ (7);"FORMAT MISTAKE, P
     LEASE REENTER ": PRINT : GOTO
     140
160  IF  ASC ( MID$ (P$,4,1)) <  >
     45 THEN  PRINT  CHR$ (7);"FO
     RMAT MISTAKE,USE'-'IN BETWEE
     N ": GOTO 120
170  HOME : FLASH : PRINT "PLEASE
      CHECK YOUR ENTRY": NORMAL
180  PRINT "NAME: ";N$
185  PRINT "PHONE NO: ";P$
190  INPUT "ARE THEY CORRECT? ";R
     $
200  IF R$ <  > "Y" AND R$ <  > "
     N" THEN  PRINT  CHR$ (7);"PL
     EASE ENTER 'Y' OR'N' ": PRINT
     : GOTO 180
210  IF R$ = "Y" THEN  HOME : GOTO
     240
220  IF R$ = "N" THEN  PRINT : PRINT
     "PLEASE REENTER NAME AND PHO
     NE NO. "
230  GOTO 100
240 M$ = N$ + P$
250  PRINT D$;"WRITE PHONE LIST2,
     R";R
260  PRINT M$
270  PRINT D$
280  INPUT "MORE ENTRIES? ";R$
285  IF  LEFT$ (R$,1) = "N" THEN
     290
288  GOTO 90
290  PRINT D$;"WRITE PHONE LIST2,
     RØ"
300  PRINT R
310  PRINT D$
320  PRINT D$;"CLOSE PHONE LIST2"
```

```
330  PRINT : PRINT  TAB( 12)"END
     OF LISTING"
340  PRINT : PRINT "NEW RECORD CO
     UNT: ";R
350  END
```

รูปที่ 6.14 โปรแกรมเพิ่มเติม RANDOM ACCESS FILE ชื่อ "ADD PHONE 2"

หลังจากที่มีการเปิด FILE แล้ว บรรทัดที่ 50 จะอ่าน FILE ที่ RECORD หมายเลข 0 ซึ่งได้บันทึกไว้แต่แรกแล้วว่าได้บันทึก RECORD ที่เท่าใดลงไปแล้ว ในตัวอย่าง เราได้ป้อนข้อมูลถึง RECORD หมายเลข 3 มาแล้ว ข้อมูลต่อมาก็จะต้องอยู่ใน RECORD หมายเลข 4 ซึ่งบรรทัดที่ 90 จะทำ 3 ให้เป็น 4 บรรทัดที่ 290-300 จะจัดไว้ใน FILE หมายเลข 0 อีกว่าได้เพิ่มเติมไปอีกตอนนี้ถึง RECORD ที่เท่าใดแล้ว หากจะมีการเพิ่มข้อมูลอีกก็ใช้โปรแกรมนี้ได้อีก

SAVE โปรแกรมนี้ไว้ในชื่อของ "ADD PHONE2" RUN โปรแกรมแล้วเพิ่มรายการนี้

CHALOTHORN	CHAYARATANA	424-6504
K.E.C.	.	277-4528
PUCHONG	UTHAYOPAS	424-0433

TEXT FILE ที่ชื่อ PHONE LIST2 จะมีรายการนี้ต่อท้าย LOAD โปรแกรมในรูปที่ 6.12 จะให้ FILE ออกมาดังนี้

```
             PHONE DIRECTORY2

KIMENG SAEPHU                      123-4567

PRATOOM PIYAKIT                    567-8900

CHAO SAEDAN                        234-6020

CHALOTHORN CHAYARATANA             424-6504

K.E.C.                             277-4528

PUTCHONG UTHYOPAS                  424-0433

              END OF FILE
```

รูปที่ 6.15 โปรแกรม PHONE DIRECTORY ที่เพิ่มเติมแล้ว

6.3.4 การแก้ไขข้อมูลใน RANDOM ACCESS FILE ทำได้โดยโปรแกรมที่แสดงในรูปที่ 6.16 โปรแกรมนี้มีสิ่งที่น่าสนใจดังนี้

บรรทัดที่ 40 จะเปิด FILE แล้วอ่านข้อมูลที่อยู่ใน RECORD หมายเลข 0 ซึ่งจะเป็นจำนวน RECORD ทั้งสิ้นที่มีอยู่ใน FILE นี้ (บรรทัดที่ 50)

บรรทัดที่ 90 เป็นการอ่านข้อมูลแต่ละ RECORD ตั้งแต่ RECORD หมายเลข 1 จนถึงหมายเลขสุดท้าย

สำหรับ FILE นี้ได้มีการนำเอาชื่อกับหมายเลขโทรศัพท์มารวมกันไว้ บรรทัดที่ 120 เป็นการแยกชื่อกับหมายเลขโทรศัพท์ออกจากกัน

บรรทัดที่ 150-210 เป็น MENU ให้เลือก

บรรทัดที่ 230-243 เป็นการสั่งให้ทำงานตามที่เลือกตาม MENU SUBROUTINE ที่ 320 เป็นการสั่งให้บันทึกข้อมูลที่เปลี่ยนแปลงไว้ในแผ่นจาน แล้วปิด FILE ในบรรทัดที่ 370

```
]LIST

10  REM  PRO#6.11
20  HOME : REM  * CHANGE PHONE LI
     ST2 *
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN PHONE LIST2,L3
     5"
50  PRINT D$;"READ PHONE LIST2,RØ
     "
60  INPUT R
70  PRINT D$
80  FOR I = 1 TO R
90  PRINT D$;"READ PHONE LIST2,R"
     ;I
100  INPUT M$
110  PRINT D$
120  HOME :N$ =  LEFT$ (M$,25):P$
      =  RIGHT$ (M$,8)
130  PRINT " NAME: ";N$
140  PRINT " PHONE NO.: ";P$
145  VTAB 8
150  PRINT "     ENTER ONE OF THES
     E OPTIONS"
160  PRINT "<1> CHANGE ALL   "
170  PRINT "<2> CHANGE NAME ONLY"

180  PRINT "<3> CHANGE PHONE NO.
     ONLY"
190  PRINT "<4> NO CHANGE"
200  PRINT
210  INPUT "     ENTER YOUR CHOIC
     E:  ";A$
220 A =  VAL (A$)
230  IF A < 1 OR A > 5 THEN  PRINT
     "ENTER 1 TO 5 ONLY": GOTO 21
     0
240  IF A = 1 THEN  GOSUB 280: GOSUB
     300: GOSUB 320: GOTO 250
241  IF A = 2 THEN  GOSUB 280: GOSUB
     320: GOTO 250
242  IF A = 3 THEN  GOSUB 300: GOSUB
     320: GOTO 250
243  IF A = 4 THEN  GOSUB 320
250  NEXT I
```

```
260  GOTO 370
270  PRINT
280  INPUT "ENTER NEW NAME: ";N$
285  IF  LEN (N$) < 25 THEN N$ =
     N$ + " ": GOTO 285
290  RETURN
300  INPUT "ENTER NEW PHONE NO.:
     ";P$
310  RETURN
320 M$ = N$ + P$
330  PRINT D$;"WRITE PHONE LIST2,
     R";I
340  PRINT M$
350  PRINT D$
360  RETURN
370  PRINT D$;"CLOSE"
380  END
```

รูปที่ 6.16 โปรแกรมแก้ไข RAMDOM ACCESS FILE ชื่อ "CHANGE PHONE LIST 2"

6.3.5 สรุปโปรแกรมเกี่ยวกับหมายเลขโทรศัพท์ที่เขียนด้วย RANDOM ACCFSS FILE เท่าที่เขียนมามีดังนี้

รูปที่	โปรแกรมชื่อ	ใช้สำหรับ
6.11	PHONE 2	สร้าง TEXT FILE ที่มีชื่อว่า "PHONE LIST"
6.12	PHONE DIRECTORY 2	อ่าน TEXT FILE ชื่อ "PHONE LIST"
6.14	ADD PHONE 2	เพิ่มเติมรายชื่อและหมายเลขโทรศัพท์ใน TEXT FILE ชื่อ "PHONE LIST"
6.16	CHANGE PHONE LIST 2	แก้ไขรายชื่อและหมายเลขโทรศัพท์ใน TEXT FILE ชื่อ "PHONE LIST"

โปรแกรมทั้ง 4 นี้เรียกว่า UTILITILY PROGRAM ต่อไปเราจะสร้าง MAIN PROGRAM โดยเอาโปรแกรมทั้ง 4 นี้มาใช้งานด้วยกัน ตัว MAIN โปรแกรมจะแสดง MENU และกำหนดว่าเมื่อใดจะโปรแกรมใด

รูปที่ 6.17 จะเป็น MAIN PROGRAM รูปที่ 6.19, 6.20, 6.21, 6.22 เป็นการอ่านโปรแกรม PHONE2, ADD PHONE2, CHANGE PHONE LIST2 และ PHONE DIRECTORY2 มาดัดแปลงเพิ่มเติมเล็กน้อย ส่วนที่เพิ่มเติมได้ขีดเส้นใต้ไว้เพื่อให้เป็นข้อสังเกต MENU ของ MAIN PROGRAM แสดงในรูปที่ 6.18

```
10  REM  PRO#6.12
20  HOME
30  REM  * PERSONAL PHONE LIST *
40  REM  * BY PRAPAT UTHAYOPAS *
50 :
60  VTAB 5
70  PRINT  TAB( 8)"PERSONAL PHONE
      LIST": PRINT : PRINT
80  PRINT "<1> CREATE PHONE DIREC
     TORY "
90  PRINT "<2> ADD THE LIST"
100  PRINT "<3> CHANGE THE LIST"
110  PRINT "<4> LIST THE DIRECTOR
     Y"
115  PRINT "<5> EXIT"
120  VTAB 20: HTAB 10
130  INPUT "YOUR CHOICE ? ";A$
140 A =  VAL (A$):D$ =  CHR$ (4)
150  IF A < 1 OR A > 5 THEN  PRINT
      CHR$ (7);"1 TO 5 PLEASE": FOR
     I = 1 TO 2000: NEXT I: GOTO
     130
160  ON A GOTO 200,300,400,500,60
     0
200  HOME : VTAB 10: HTAB 4
```

```
210   PRINT "YOUR PREVIOUS PHONE D
      IRECTORY WILL BE COMPLETELY
      ERASED": PRINT
220   INPUT "PLEASE CONFIRM 'Y' OR
      'N' ";A$
230   IF A$ = "N" THEN 70
250   PRINT D$;"RUN PHONE2"
300   PRINT D$;"RUN ADD PHONE LIST
      2"
400   PRINT D$;"RUN CHANGE PHONE L
      IST2"
500   PRINT D$;"RUN PHONE DIRECTOR
      Y2"
600   HOME : VTAB 10: PRINT  TAB(
      16)"GOOD BYE": FOR I = 1 TO
      2000: NEXT I: HOME : END
```

รูปที่ 6.17 MAIN PROGRAM ชื่อ PERSONAL PHONE LIST

```
        PERSONAL PHONE LIST


<1> CREATE PHONE DIRECTORY
<2> ADD THE LIST
<3> CHANGE THE LIST
<4> LIST THE DIRECTORY
<5> EXIT




          YOUR CHOICE ? 1
   YOUR PREVIOUS PHONE DIRECTORY WILL BE
    COMPLETELY ERASED

PLEASE CONFIRM 'Y' OR 'N' Y


TYPE 'STOP' IF NO MORE DATA

ENTER NAME (NOT MORE THAN 25 CHARACTERS)
 KIMENG SAEPHU
```

```
PHONE NUMBER FORMAT: XXX-XXXX
ENTER THE NUMBER:    123-4567
PLEASE'CHECK'YOUR'ENTRY
NAME: KIMENG SAEPHU
PHONE NO.: 123-4567
ARE THEY CORRECT? Y
TYPE 'STOP' IF NO MORE DATA

ENTER NAME (NOT MORE THAN 25 CHARACTERS)
 PRATOOM PIYAKIT

PHONE NUMBER FORMAT: XXX-XXXX
ENTER THE NUMBER:    567-8900
PLEASE'CHECK'YOUR'ENTRY
NAME: PRATOOM PIYAKIT
PHONE NO.: 567-8900
ARE THEY CORRECT? Y
TYPE 'STOP' IF NO MORE DATA

ENTER NAME (NOT MORE THAN 25 CHARACTERS)
 CHAO SAEDAN

PHONE NUMBER FORMAT: XXX-XXXX
ENTER THE NUMBER:    234-6020
PLEASE'CHECK'YOUR'ENTRY
NAME: CHAO SAEDAN
PHONE NO.: 234-6020
ARE THEY CORRECT? Y
TYPE 'STOP' IF NO MORE DATA

ENTER NAME (NOT MORE THAN 25 CHARACTERS)STOP

          END OF LISTING

        PERSONAL PHONE LIST



<1> CREATE PHONE DIRECTORY
<2> ADD THE LIST
<3> CHANGE THE LIST
<4> LIST THE DIRECTORY
<5> EXIT
        YOUR CHOICE ? 2
```

```
RECORD NO: 3

ENTER NAME (NOT MORE THAN 25 CHARACTERS):
 CHALOTHORN CHAIYA

PHONE NUMBER FORMAT: XXX-XXXX
ENTER THE NUMBER:    424-6504
PLEASE'CHECK'YOUR'ENTRY
NAME: CHALOTHORN CHAIYA
PHONE NO: 424-6504
ARE THEY CORRECT? Y
MORE ENTRIES? Y
ENTER NAME (NOT MORE THAN 25 CHARACTERS):
 K.E.C.

PHONE NUMBER FORMAT: XXX-XXXX
ENTER THE NUMBER:    277-0149
PLEASE'CHECK'YOUR'ENTRY
NAME: K.E.C.
PHONE NO: 277-0149
ARE THEY CORRECT? Y
MORE ENTRIES? Y
ENTER NAME (NOT MORE THAN 25 CHARACTERS):
 PUTCHONG UTHAYOPAS

PHONE NUMBER FORMAT: XXX-XXXX
ENTER THE NUMBER:    424-0433
PLEASE'CHECK'YOUR'ENTRY
NAME: PUTCHONG UTHAYOPAS
PHONE NO: 424-0433
ARE THEY CORRECT? Y
MORE ENTRIES? N

           END OF LISTING

NEW RECORD COUNT: 6

        PERSONAL PHONE LIST


<1> CREATE PHONE DIRECTORY
<2> ADD THE LIST
<3> CHANGE THE LIST
<4> LIST THE DIRECTORY
<5> EXIT
```

```
        YOUR CHOICE ? 3

 NAME: KIMENG SAEPHU
 PHONE NO.: 123-4567
    ENTER ONE OF THESE OPTIONS
<1> CHANGE ALL
<2> CHANGE NAME ONLY
<3> CHANGE PHONE NO. ONLY
<4> NO CHANGE

     ENTER YOUR CHOICE:  1
ENTER NEW NAME: RATANAVADEE PANICHKULTHORN
ENTER NEW PHONE NO.: 123-4566
 NAME: PRATOOM PIYAKIT
 PHONE NO.: 567-8900
    ENTER ONE OF THESE OPTIONS
<1> CHANGE ALL
<2> CHANGE NAME ONLY
<3> CHANGE PHONE NO. ONLY
<4> NO CHANGE

     ENTER YOUR CHOICE:  2
ENTER NEW NAME: TAVEESAK PIYAKIT
 NAME: CHAO SAEDAN
 PHONE NO.: 234-6020
    ENTER ONE OF THESE OPTIONS
<1> CHANGE ALL
<2> CHANGE NAME ONLY
<3> CHANGE PHONE NO. ONLY
<4> NO CHANGE

     ENTER YOUR CHOICE:  3
ENTER NEW PHONE NO.: 286-1981
 NAME: CHALOTHORN CHAIYA
 PHONE NO.: 424-6504
    ENTER ONE OF THESE OPTIONS
<1> CHANGE ALL
<2> CHANGE NAME ONLY
<3> CHANGE PHONE NO. ONLY
<4> NO CHANGE

     ENTER YOUR CHOICE:  4
 NAME: K.E.C.
 PHONE NO.: 277-0149
    ENTER ONE OF THESE OPTIONS
```

```
<1> CHANGE ALL
<2> CHANGE NAME ONLY
<3> CHANGE PHONE NO. ONLY
<4> NO CHANGE

     ENTER YOUR CHOICE:  4
 NAME: PUTCHONG UTHAYOPAS
 PHONE NO.: 424-0433
    ENTER ONE OF THESE OPTIONS
<1> CHANGE ALL
<2> CHANGE NAME ONLY
<3> CHANGE PHONE NO. ONLY
<4> NO CHANGE

     ENTER YOUR CHOICE:  4

       PERSONAL PHONE LIST


<1> CREATE PHONE DIRECTORY
<2> ADD THE LIST
<3> CHANGE THE LIST
<4> LIST THE DIRECTORY
<5> EXIT
         YOUR CHOICE ? 4

           PHONE DIRECTORY2

RATANAVADEE PANICHKULTHORN123-4566

TAVEESAK PIYAKIT          567-8900

CHAO SAEDAN              286-1981

CHALOTHORN CHAIYA         424-6504

K.E.C.                   277-0149

PUTCHONG UTHAYOPAS        424-0433

           END OF FILE

PRESS 'RETURN ' TO GO BACK TO THE MAIN MENU

       PERSONAL PHONE LIST
```

```
<1> CREATE PHONE DIRECTORY
<2> ADD THE LIST
<3> CHANGE THE LIST
<4> LIST THE DIRECTORY
<5> EXIT
         YOUR CHOICE ? 5
               GOOD BYE
```

รูปที่ 6.18 MENU และการใช้งานของ MAIN PROGRAM ชื่อ PERSONAL PHONE LIST

```
10  REM  PRO#6.8
20  HOME : REM  *PHONE2*
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN PHONE LIST2"
50  PRINT D$;"DELETE PHONE LIST2"

60  PRINT D$;"OPEN PHONE LIST2,L3
    5"
70 R = 0
80  PRINT
90  PRINT "TYPE 'STOP' IF NO MORE
     DATA": PRINT
100  INPUT "ENTER NAME (NOT MORE
     THAN 25 CHARACTERS)";N$
110  IF N$ = "STOP" THEN 320
120 R = R + 1
130  IF  LEN (N$) > 25 THEN  PRINT
      CHR$ (7);"NOT MORE THAN 25
     CHAR. PLEASE": PRINT : GOTO
     100
140  IF  LEN (N$) < 25 THEN N$ =
     N$ + " ": GOTO 140
145  PRINT
150  PRINT "PHONE NUMBER FORMAT:
     XXX-XXXX"
160  INPUT "ENTER THE NUMBER:
     ";P$
170  IF  LEN (P$) <  > 8 THEN  PRINT
      CHR$ (7);"FORMAT MISTAKE, P
     LEASE REENTER": PRINT : GOTO
     160
```

```
180  IF  ASC ( MID$ (P$,4,1)) <  >
     45 THEN  PRINT  CHR$ (7);"FO
     RMAT MISTAKE, USE'-' IN BETW
     EEN": GOTO 160
190  HOME : FLASH : PRINT "PLEASE
      CHECK YOUR ENTRY": NORMAL
200  PRINT "NAME: ";N$
210  PRINT "PHONE NO.: ";P$
220  INPUT "ARE THEY CORRECT? ";R
     $
230  IF R$ <  > "Y" AND R$ <  > "
     N" THEN  PRINT  CHR$ (7);"PL
     EASE ENTER 'Y' OR 'N' ": PRINT
     : GOTO 220
240  IF R$ = "Y" THEN  HOME : GOTO
     270
250  IF R$ = "N" THEN  PRINT : PRINT
     "PLEASE REENTER NAME AND PHO
     NE NO."
260  GOTO 80
270 M$ = N$ + P$
280  PRINT D$;"WRITE PHONE LIST2,
     R";R
290  PRINT M$
300  PRINT D$
310  GOTO 90
320  PRINT D$;"WRITE PHONE LIST2,
     R0"
330  PRINT R
340  PRINT D$
350  PRINT D$;"CLOSE PHONE LIST2"

360  PRINT : PRINT  TAB( 12)"END
     OF LISTING"
370  PRINT D$;"RUN PERSONAL PHONE
      LIST"
```

รูปที่ 6.19 UTILITY PROGRAM ชื่อ PHONE 2

```
10  REM  PRO#6.10
20  HOME : REM  * ADD PHONE2 *
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN PHONE LIST2,L3
    5"
```

```
50  PRINT D$;"READ PHONE LIST2,RO
    "
60  INPUT R
70  PRINT D$
80  PRINT : PRINT "RECORD NO: ";R
    : PRINT
90 R = R + 1
100  INPUT "ENTER NAME (NOT MORE
     THAN 25 CHARACTERS):";N$
110  IF  LEN (N$) > 25 THEN  PRINT
      CHR$ (7);"NOT MORE THAN 25
     CHAR.,PLEASE": GOTO 100
120  IF  LEN (N$) < 25 THEN N$ =
     N$ + " ": GOTO 120
125  PRINT
130  PRINT "PHONE NUMBER FORMAT:
     XXX-XXXX"
140  INPUT "ENTER THE NUMBER:
     ";P$
150  IF  LEN (P$) <  > 8 THEN  PRINT
      CHR$ (7);"FORMAT MISTAKE, P
     LEASE REENTER ": PRINT : GOTO
     140
160  IF  ASC ( MID$ (P$,4,1)) <  >
     45 THEN  PRINT  CHR$ (7);"FO
     RMAT MISTAKE,USE'-'IN BETWEE
     N ": GOTO 120
170  HOME : FLASH : PRINT "PLEASE
      CHECK YOUR ENTRY": NORMAL
180  PRINT "NAME: ";N$
185  PRINT "PHONE NO: ";P$
190  INPUT "ARE THEY CORRECT? ";R
     $
200  IF R$ <  > "Y" AND R$ <  > "
     N" THEN  PRINT  CHR$ (7);"PL
     EASE ENTER 'Y' OR'N' ": PRINT
     : GOTO 180
210  IF R$ = "Y" THEN  HOME : GOTO
     240
220  IF R$ = "N" THEN  PRINT : PRINT
     "PLEASE REENTER NAME AND PHO
     NE NO. "
230  GOTO 100
240 M$ = N$ + P$
250  PRINT D$;"WRITE PHONE LIST2,
     R";R
```

```
260  PRINT M$
270  PRINT D$
280  INPUT "MORE ENTRIES? ";R$
285  IF  LEFT$ (R$,1) = "N" THEN
     290
288  GOTO 90
290  PRINT D$;"WRITE PHONE LIST2,
     R0"
300  PRINT R
310  PRINT D$
320  PRINT D$;"CLOSE PHONE LIST2"

330  PRINT : PRINT  TAB( 12)"END
     OF LISTING"
340  PRINT : PRINT "NEW RECORD CO
     UNT: ";R
350  PRINT D$;"RUN PERSONAL PHONE
      LIST"
```

รูปที่ 6.20 UTILITY PROGRAM ชื่อ ADD PHONE 2

```
10  REM  PRO#6.11
20  HOME : REM  * CHANGE PHONE LI
    ST2 *
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN PHONE LIST2,L3
    5"
50  PRINT D$;"READ PHONE LIST2,R0
    "
60  INPUT R
70  PRINT D$
80  FOR I = 1 TO R
90  PRINT D$;"READ PHONE LIST2,R"
    ;I
100  INPUT M$
110  PRINT D$
120  HOME :N$ =  LEFT$ (M$,25):P$
     =  RIGHT$ (M$,8)
130  PRINT " NAME: ";N$
140  PRINT " PHONE NO.: ";P$
145  VTAB 8
```

```
150  PRINT "     ENTER ONE OF THES
     E OPTIONS"
160  PRINT "<1> CHANGE ALL   "
170  PRINT "<2> CHANGE NAME ONLY"

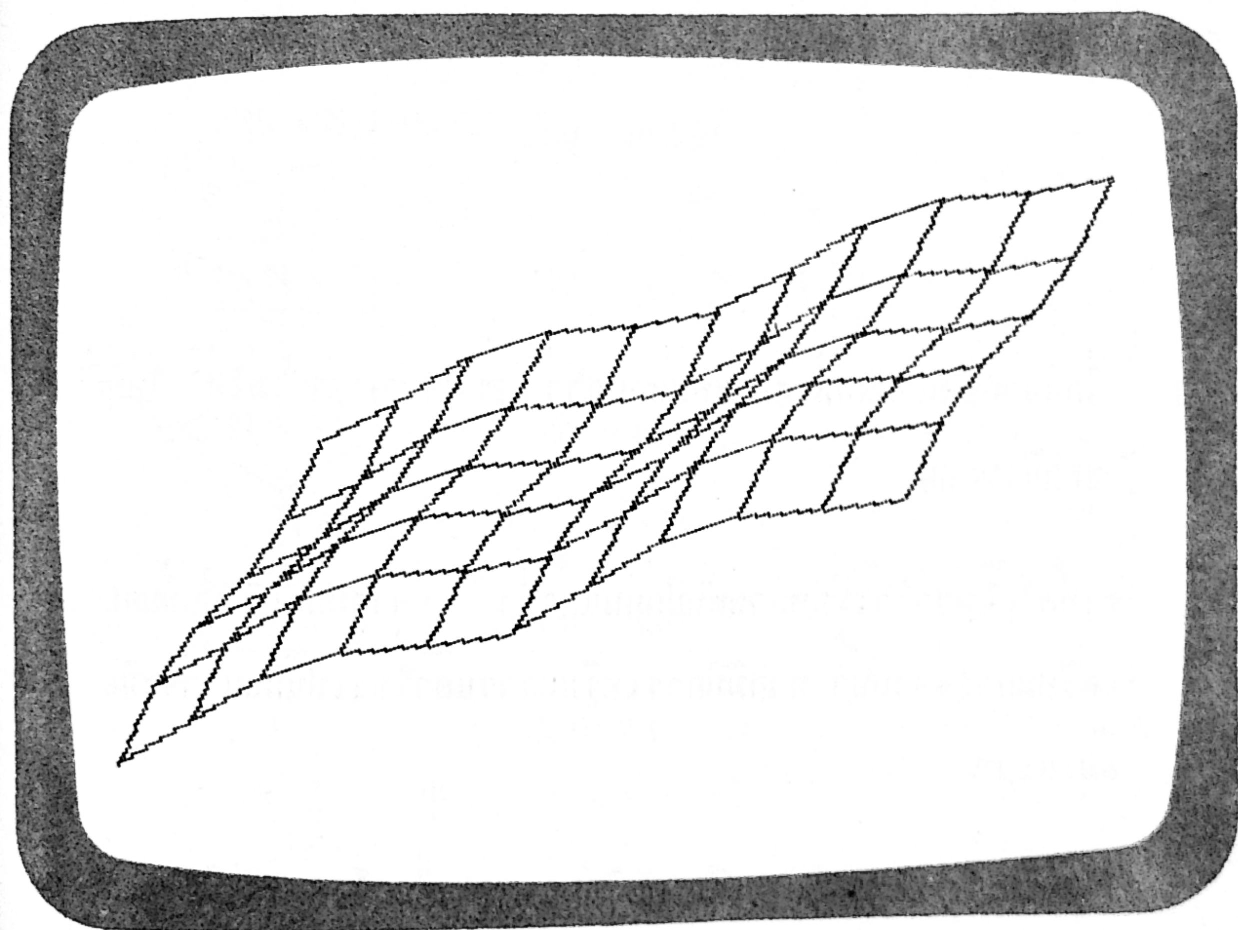
180  PRINT "<3> CHANGE PHONE NO.
     ONLY"
190  PRINT "<4> NO CHANGE"
200  PRINT
210  INPUT "      ENTER YOUR CHOIC
     E:   ";A$
220 A =  VAL (A$)
230  IF A < 1 OR A > 5 THEN  PRINT
     "ENTER 1 TO 5 ONLY": GOTO 21
     0
240  IF A = 1 THEN  GOSUB 280: GOSUB
     300: GOSUB 320: GOTO 250
241  IF A = 2 THEN  GOSUB 280: GOSUB
     320: GOTO 250
242  IF A = 3 THEN  GOSUB 300: GOSUB
     320: GOTO 250
243  IF A = 4 THEN  GOSUB 320
250  NEXT I
260  GOTO 370
270  PRINT
280  INPUT "ENTER NEW NAME:  ";N$
285  IF  LEN (N$) < 25 THEN N$ =
     N$ + " ": GOTO 285
290  RETURN
300  INPUT "ENTER NEW PHONE NO.:
     ";P$
310  RETURN
320 M$ = N$ + P$
330  PRINT D$;"WRITE PHONE LIST2,
     R";I
340  PRINT M$
350  PRINT D$
360  RETURN
370  PRINT D$;"CLOSE"
380  PRINT D$;"RUN PERSONAL PHONE
      LIST"
```

รูปที่ 6.21 UTILITY PROGRAM ชื่อ CHANGE PHONE LIST 2

```
10  REM   PRO#6.9
20  HOME : PRINT  TAB( 12);"PHONE
      DIRECTORY2": PRINT
30 D$ =  CHR$ (4)
40  PRINT D$;"OPEN PHONE LIST2,L3
     5"
50  PRINT D$;"READ PHONE LIST2,R0
     "
60  INPUT R
70  PRINT D$
80  FOR I = 1 TO R
90  PRINT D$;"READ PHONE LIST2,R"
     I
100  INPUT M$
110  PRINT D$
120  PRINT M$: PRINT
130  NEXT I
140  PRINT D$;"CLOSE"
150  PRINT  TAB( 12)"END OF FILE"

155  PRINT : INPUT "PRESS 'RETURN
      ' TO GO BACK TO THE MAIN ME
     NU";A$
160  PRINT D$;"RUN PERSONAL PHONE
      LIST"
```

รูปที่ 6.22 UTILITY PROGRAM ชื่อ PHONE DIRECTORY 2



7

FLOW CHART

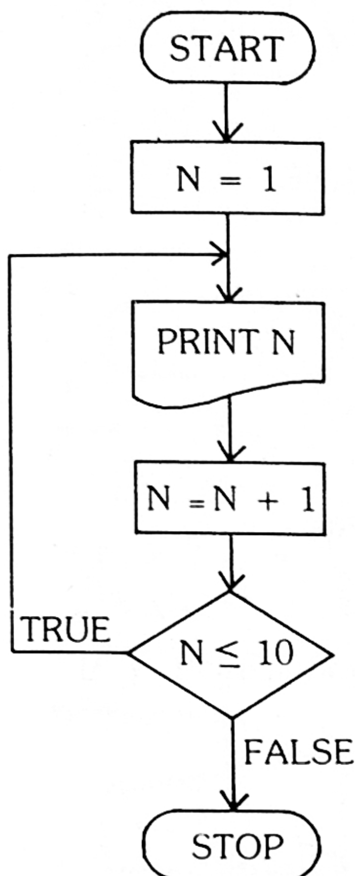
7.1 เนื่องจากยังหาศัพท์ที่เหมาะสมแทนคำว่า FLOW CHART ไม่ได้ ในที่นี้จึงขอใช้คำนี้ไปก่อน

ขงจื้อได้กล่าวไว้เมื่อหลายพันปีก่อนแล้วว่า "ความสำเร็จนั้นขึ้นอยู่กับการเตรียมการล่วงหน้า หากมิมีการเตรียมการแล้วไซร้เป็นที่แน่ว่าจะมีแต่ความล้มเหลว"

การเขียนโปรแกรมก็เช่นเดียวกับกิจการอย่างอื่น ซึ่งควรจะมีการเตรียมตัวล่วงหน้า FLOW CHART เปรียบเสมือนแบบร่างเค้าโครงของโปรแกรมที่เราจะเขียน ช่วยไม่ให้หลงลืมขั้นตอนบางขั้น และเป็นเสมือนการสื่อสารในระหว่างบุคคลในกลุ่มที่ทำงานในโครงการเดียวกัน ผู้ที่เริ่มต้นเขียนโปรแกรมหากฝึกให้มีการเขียน FLOW CHART ก่อนให้เป็นนิสัยแล้วต่อไปก็จะเป็นนักเขียนโปรแกรมที่ดี

ดูไดอะแกรมในรูปที่ 7.1

FLOW CHART　　　　　　　　โปรแกรม

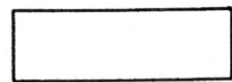



```
10   REM  PROGRAM TO PRINT  1 TO 10
.20  N =  1
30   PRINT  N
40   N = N + 1
50   IF N <=  10  THEN  30
60   END
```

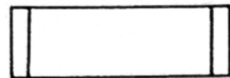
รูปที่ 7.1

รูปที่ 7.1 แสดง FLOW CHART เทียบกับโปรแกรมง่ายๆ ที่ให้พิมพ์ตัวเลขตั้งแต่ 1 ถึง 10 จะเห็นว่าหากดูจาก FLOW CHART จะเห็นได้ง่ายกว่าดูจากโปรแกรมเอง ความจริงแล้ว FLOWCHART ก็คือเส้นทางเดินของโปรแกรมนั่นเอง

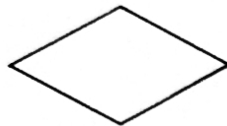
7.2　สัญลักษณ์ของส่วนประกอบของ FLOW CHART สถาบันมาตรฐานแห่งชาติของสหรัฐ(AMERICAN NATIONAL STANDARDS INSTITUTE-ANSI) ได้กำหนดสัญลักษณ์เพื่อเขียน FLOW CHART ไว้ดังนี้



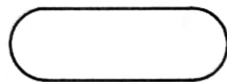
กรรมวิธีทั่ว ๆ ไป



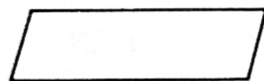
ไปที่ SUBROUTINE



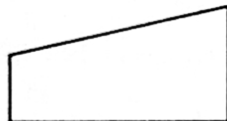
ทดสอบหรือตัดสิน



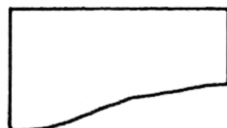
จุดเริ่ม หรือ จุดสิ้นสุด



การป้อนหรือการแสดงข้อมูลทั่ว ๆ ไป



การป้อนข้อมูลทางแป้นพิมพ์

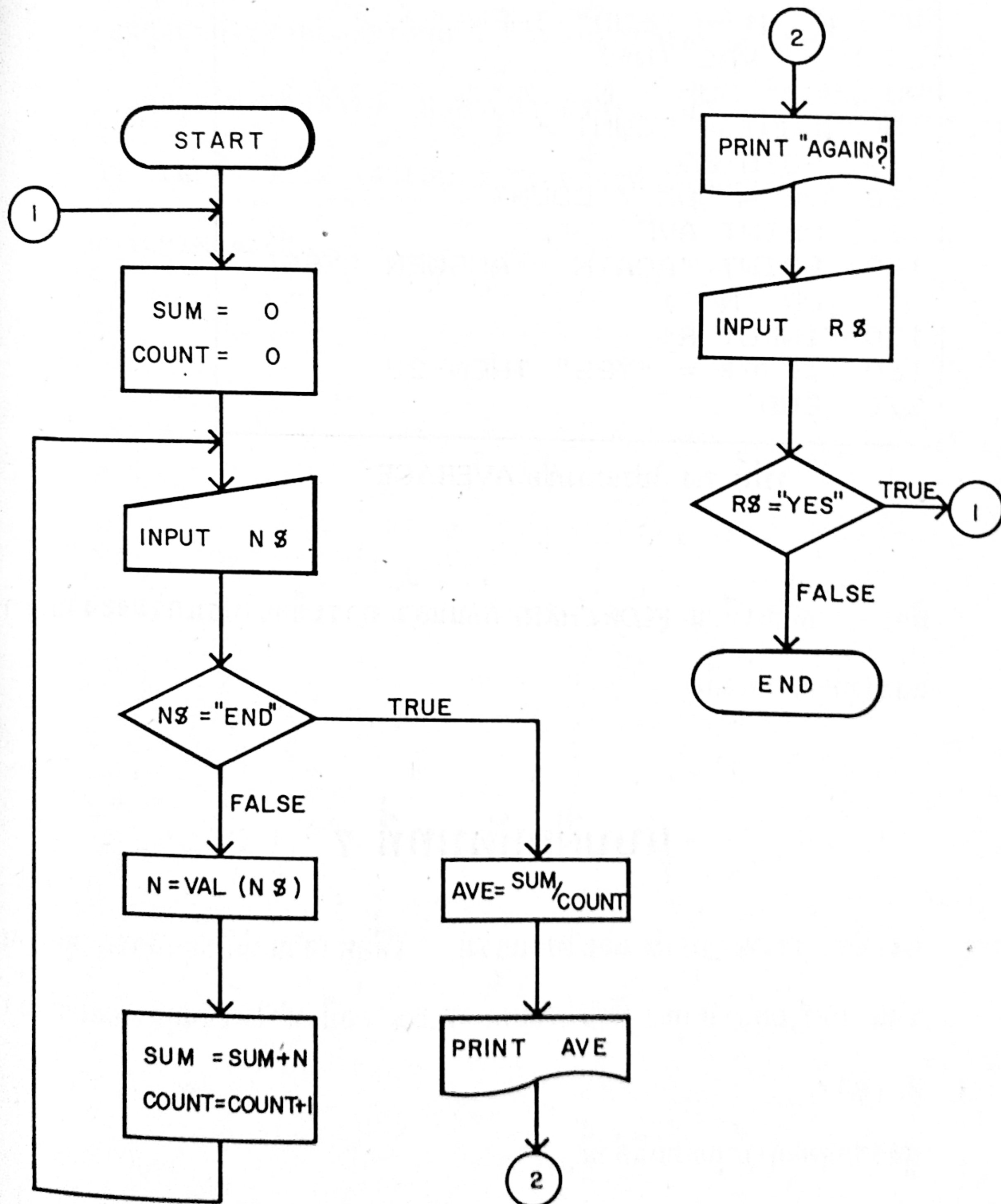


การแสดงข้อมูลทางเครื่องพิมพ์

รูปที่ 7.2 สัญลักษณ์ของ FLOW CHART

7.3 ต่อไปนี้เป็นตัวอย่างของการเขียน FLOWCHART ของโปรแกรม โจทย์มีว่า ให้หาผลเฉลี่ยของตัวเลขกลุ่มหนึ่งที่ไม่จำกัดจำนวน ซึ่งถ้าหากว่าตัวเลขกลุ่มนั้นหมดแล้วให้พิมพ์คำว่า "END" ลงไป แล้วเครื่องจะบวกตัวเลขกลุ่มนั้นแล้วหาผลเฉลี่ยแสดงออกมาทางจอภาพ(หรือเครื่องพิมพ์) จากนี้เครื่องจะ

ถามว่าจะมีการกระทำเช่นนี้อีกหรือไม่ หากมีก็ให้ย้อนกลับไปต้นโปรแกรมใหม่ หากไม่มีก็หมดโปรแกรม รูปที่ 7.3 เป็น FLOW CHART และรูปที่ 7.4 เป็นโปรแกรม



รูปที่ 7.3 FLOW CHART ของโปรแกรมชื่อ AVERAGE

```
10  REM  AVERAGE
20  HOME
30 SUM = 0
40 COUNT = 0
50  PRINT "ENTER'END'WHEN FINISH"

60  INPUT N$
70  IF N$ = "END" THEN 120
80 N =  VAL (N$)
90 SUM = SUM + N
100 COUNT = COUNT + 1
110  GOTO 60
120 AVE = SUM / COUNT
130  PRINT AVE
140  PRINT "AGAIN ?(ANSWER 'YES'
     OR 'NO')
150  INPUT R$
160  IF R$ = "YES" THEN 20
170  END
```

รูปที่ 7.4 โปรแกรมชื่อ AVERAGE

สังเกตว่าหากเขียน FLOW CHART ก่อนแล้ว การเขียนโปรแกรมจะง่ายเข้า และไม่มีการหลงลืม

แบบฝึกหัดบทที่ 7

7.1 จงเขียน FLOW CHART และโปรแกรม เพื่อหาจำนวนปีของการลงทุนด้วยเงิน 100,000 บาท มีกำไรทบต้น 7.5% ต่อปี ทำให้เงินทุนเพิ่มมากกว่า 2 เท่า

สูตรของดอกเบี้ยทบต้นมีดังนี้

$$A = P(1+R)^N$$

P คือเงินลงทุน 100,000 บาท

R คืออัตราดอกเบี้ย 7.5% = 0.075

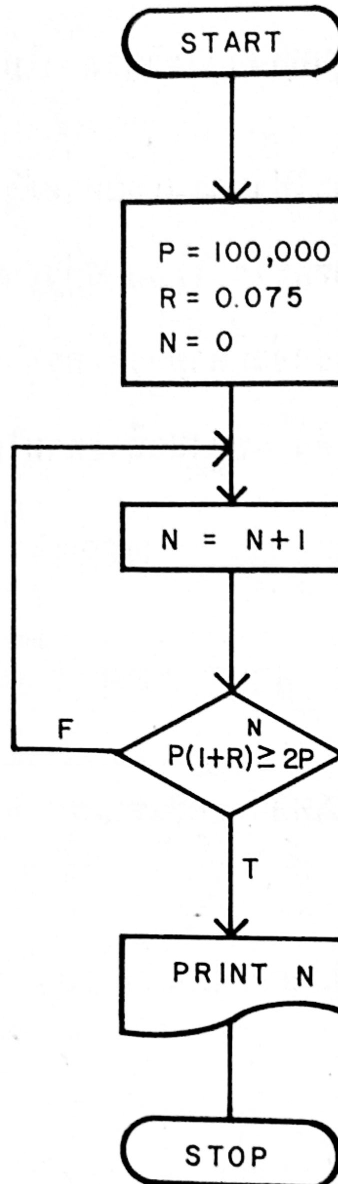
N คือจำนวนปี

A คือเงินลงทุนหลังจากเวลาล่วงไป N ปี

7.2 จงเขียน FLOW CHART และโปรแกรมเกมทายตัวเลข โดยเครื่องจะถามว่าเลขสูงสุดที่จะทายได้คือเลขอะไร เมื่อป้อนตัวเลขที่ทายเข้าไปเครื่องจะบอกว่าสูงไป หรือต่ำไป ถ้าตัวที่ทายตรงเครื่องจะแสดงความยินดีแล้วถามว่าจะเล่นอีกหรือไม่ ถ้าเล่นเลขสูงสุดที่จะทายได้คือเลขอะไร ถ้าไม่เล่นโปรแกรมก็จะจบ

เฉลยแบบฝึกหัดบทที่ 7

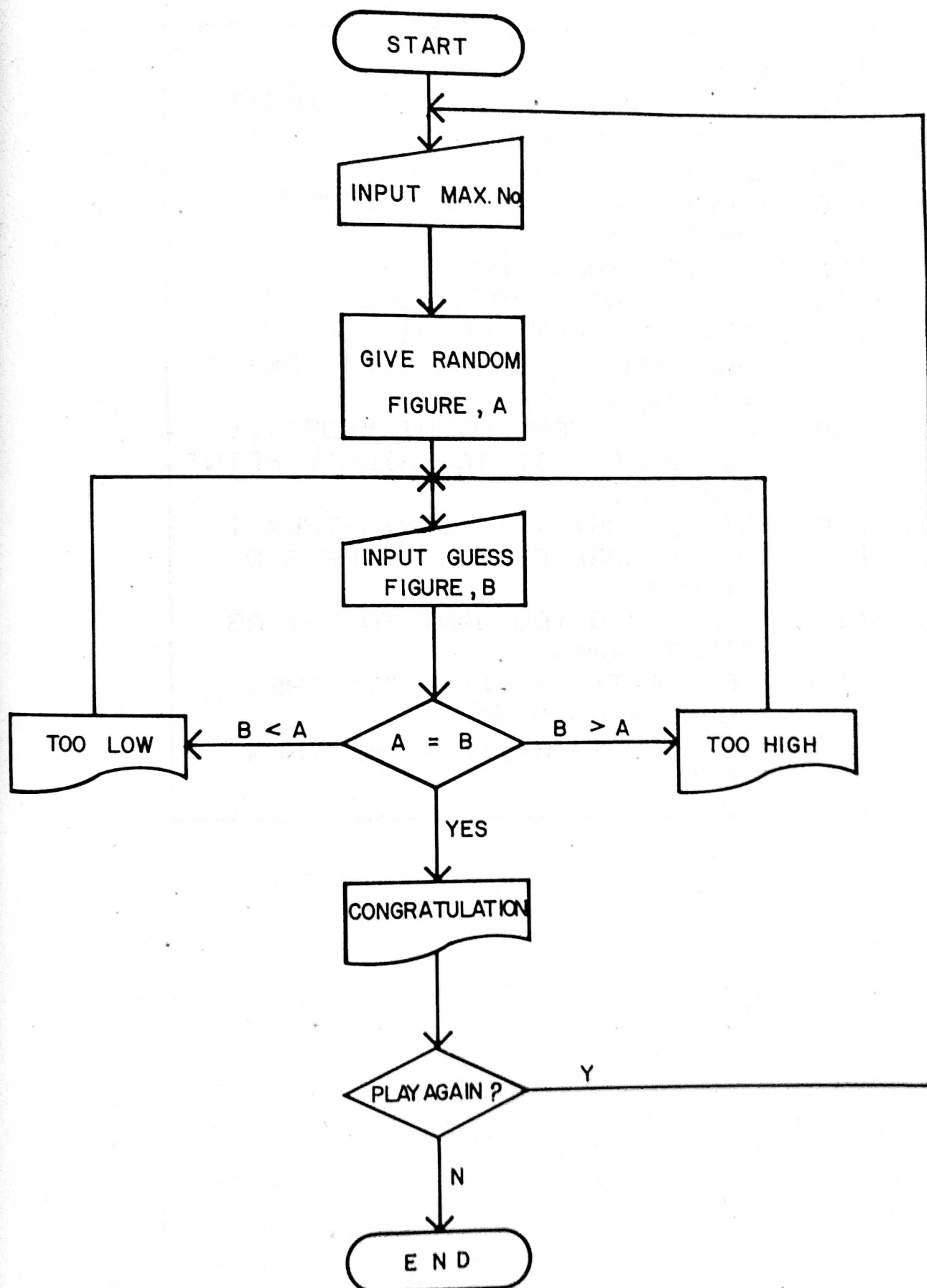
7.1 FLOW CHART



โปรแกรม

```
5  REM  SOL#7.1
10 P = 100000
20 R = .075
25 N = 0
30 N = N + 1
40  IF P * (1 + R) ^ N < 2 * P THEN
    30
50  PRINT "NO.OF YEAR ";N
60  END
```

7.2 FLOW CHART



```
10  REM  SOL#7.2
20  HOME : PRINT  TAB( 10)"NUMBER
      GUESSING GAME"
30  PRINT : PRINT
40  INPUT "WHAT IS THE MAXIMUM NU
     MBER ";M
50 A =  INT (M *  RND (1)) + 1
60  INPUT "ENTER YOUR GUESS ";B
70  IF A > B THEN  PRINT "SORRY,Y
     OUR NUMBER IS TOO LOW": PRINT
     : GOTO 60
80  IF A < B THEN  PRINT "SORRY,Y
     OUR NUMBER IS TOO HIGH": PRINT
     : GOTO 60
90  PRINT  CHR$ (7);"CONGRATULATI
     ON,YOU ARE RIGHT ON THE SPOT
     ": PRINT
100  INPUT "DO YOU WANT TO TRY AG
     AIN ? ";R$
110  IF  LEFT$ (R$,1) = "Y" THEN
      HOME : GOTO 40
120  IF  LEFT$ (R$,1) = "N" THEN
      END
```

ภาคผนวก

รหัสแอสกี้	ตัวที่แสดงบนจอภาพ	ตัวบนแป้นพิมพ์
0		ctrl @
1		ctrl A
2		ctrl B
3		ctrl C
4		ctrl D
5		ctrl E
6		ctrl F
7		ctrl G
8		ctrl H or ←
9		ctrl I
10		ctrl J
11		ctrl K
12		ctrl L
13		ctrl M or RETURN
14		ctrl N
15		ctrl O
16		ctrl P
17		ctrl Q
18		ctrl R
19		ctrl S
20		ctrl T
21		ctrl U or →
22		ctrl V
23		ctrl W
24		ctrl X
25		ctrl Y
26		ctrl Z
27		ESC
28		
29		ctrl shift-M
30		ctrl ^
31		
32	SPACE	space
33	!	!
34	"	"
35	#	#
36	$	$
37	%	%
38	&	&
39	’	’
40	(	(
41	)	)
42	*	*
43	+	+
44	,	,
45	-	-
46	.	.
47	/	/
48	0	0

รหัสแอสกี้	ตัวที่แสดงบนจอภาพ	ตัวบนแป้นพิมพ์
49	1	1
50	2	2
51	3	3
52	4	4
53	5	5
54	6	6
55	7	7
56	8	8
57	9	9
58	:	:
59	;	;
60	<	<
61	=	=
62	>	>
63	?	?
64	@	@
65	A	A
66	B	B
67	C	C
68	D	D
69	E	E
70	F	F
71	G	G
72	H	H
73	I	I
74	J	J
75	K	K
76	L	L
77	M	M
78	N	N
79	O	O
80	P	P
81	Q	Q
82	R	R
83	S	S
84	T	T
85	U	U
86	V	V
87	W	W
88	X	X
89	Y	Y
90	Z	Z
91	[	
92	\	
93	]	] (shift-M)
94	^	^
95	_	

หนังสืออ้างอิง

1. APPLE II USER'S GUIDE. -LON POOLE WITH MARTIN MC. NIFF & STEVEN COOK
2. APPLE BASIC : DATA FILE PROGRAMMING. -LE ROY FINKEL AND JERALD R. BROWN
3. APPLE BASIC. -RICHARD HASKELL
4. APPLESOFT TUTORIAL. -APPLE COMPUTER INC.
5. BASIC PROGRAMMING REFERENCE MANUAL. -APPLE COMPUTER INC.
6. BASIC. -ROBERT J. BENT & GEORGE C. SETHARES
7. THE DOS MANUAL. -APPLE COMPUTER INC.

ดัชนี

ABS( )	37
ALGOL	16
ANSI	129
APL	16
ARGUMENT	40
ASC(" ")	38
ASCII CODE	20
ATN( )	36
BASIC	16
BASIC	17
BINARY DIGIT	12
BIT	12
BOOT DISKETTE	28
BUILT-IN FUNCTIONS	35
BYTE	12
CATHODE RAY TUBE	9
CHR$( )	20
CHR$( )	38
CLOSE	91
COBOL	16
COLOR=	52
COS( )	35
CPU-CENTRAL PROCESSOR UNIT	9
CRT	9
CRT	48
DERIVED FUNCTION	40
DISK	9
DISK DRIVE	9
DISKETTE	9
DRAW	75
ERROR	30
EXP( )	38
FLOPPY DISK	14
FLOW CHART	128
FORTRAN	16
FULL SCREEN GRAPHICS	48
FUNCTIONS	34

FUNCTIONS,BUILT-IN	34
FUNCTIONS,DERIVED	34
FUNCTIONS,ELEMENTARY	34
FUNCTIONS,LIBRARY	34
FUNCTIONS,NUMERIC	34
FUNCTIONS,STANDARD	34
FUNCTIONS,STRING	34
FUNCTIONS,SYSTEM	34
FUNCTIONS,USER DEFINED	34
GR	51
GRAPHICS MODE	48
GRAPHICS,HIGH RESOLUTION	48
GRAPHICS,LOW RESOLUTION	48
HARD DISK	14
HARDWARE	8
HCOLOR=	59
HELLO	27
HEXADECIMAL	70
HGR	57
HGR2	57
HLIN	54
HPLOT	59
IF..THEN	23
INITIALIZE	28
INPUT UNIT	8
INPUT UNIT	9
INT( )	36
INTERNAL MEMORY	9
K BYTE	13
KB	13
KEYBOARD	9
KILOBYTE	13
LEFT$( )	39
LEN( )	39
LOG( )	38
LOG( )	38
LOW RESOLUTION GRAPHICS	48
MASS MEMORY	9
MEMORY UNIT	8
MID$( )	39
MONITOR	9
MONITOR	48
NO	31
NUMERIC FUNCTION	35
ONERR GOSUB...	30
ONERR GOTO...	30

OPEN 90
OUTPUT UNIT 8
OUTPUT UNIT 9
PASCAL 16
PL/1 16
PLOT 53
PRINT 90
PRINT CHR$(4) 27
PRINTER 9
PROCESSOR UNIT 8
RAM 10
RANDOM ACCESS FILE 89
RANDOM ACCESS FILE 104
RANDOM ACCESS MEMORY 10
RANDOM ACCESS MEMORY 10
READ 91
READ ONLY MEMORY 10
RECORD 104
RIGHT$( ) 39
RND( ) 36
ROM 10
ROT 74
RPG 1[illegible]
SCALE= 74
SCIENTIFIC NOTATION 26
SEQUENTIAL FILE 87
SEQUENTIAL TEXT FILE 90
SGN( ) 37
SHAPE TABLE 70
SIN( ) 35
SOFTWARE 15
SOFTWARE,APPLICATION 15
SOFTWARE,SYSTEM 15
SPC( ) 25
SQR( ) 38
STORAGE UNIT 8
STR$ 21
STR$( ) 39
STRING FUNCTION 38
SYSTEM FUNCTION 35
TAB( ) 25
TAN( ) 35
TEXT FILE 88
TEXT MODE 48
TEXT WINDOW 49
USER DEFINED FUCTION 40

VAL 21
VAL( ) 39
VLIN 55
WRITE 90
XDRAW 75
YES 31

4 ขั้นตอนเมื่อเราเริ่มเล่นแอปเปิ้ลII









เรียน APPLESOFT BASIC ด้วยตนเอง

หนังสือเล่มนี้จะช่วยให้ท่านสามารถสื่อสารหรือเขียนโปรแกรมสั่งงานคอมพิวเตอร์ได้ด้วยตัวท่านเอง โดยจะเริ่มตั้งแต่คำถามแรกที่ทุกคนต้องถามนั่นคือ คอมพิวเตอร์คืออะไร ติดตามด้วยระบบคอมพิวเตอร์เป็นอย่างไร มีอะไรบ้าง จะใช้งานคอมพิวเตอร์ต้องสั่งงานอย่างไร คำสั่งที่ใช้มีอะไรบ้าง ตลอดเล่มจะมีโปรแกรมที่ทำการทดสอบแล้วโดยผ่านเครื่อง APPLE II ให้ท่านฝึกโดยเริ่มจากง่าย ๆ และเพิ่มความยากขึ้นเรื่อย ๆ ตามคำสั่งที่ท่านได้เรียนรู้ในแต่ละบท จนในที่สุดท่านก็จะสามารถอ่านโปรแกรมเข้าใจได้ด้วยตัวท่านเอง ความหนา 170 หน้า พิมพ์ด้วยกระดาษปอนด์อย่างดี *ราคาเล่มละ 45 บาท*

เรียน/เล่น/ใช้โปรแกรม APPLESOFT BASIC

เมื่อท่านสามารถอ่านโปรแกรมเข้าใจแล้ว หนังสือเล่มนี้จะเป็นขั้นตอนต่อไปที่ช่วยเพิ่มประสบการณ์แก่ท่าน ในด้านการหัดป้อนข้อมูล (KEY) หัด SAVE, LOAD, LIST และ RUN หัดแก้และหัดแปลงโปรแกรมซึ่งต่อไปก็สามารถจะเรียนโปรแกรมเองได้ สำหรับโปรแกรม 40 โปรแกรมที่เราได้รวบรวมไว้ในหนังสือเล่มนี้ มีทั้งที่เป็นเกม โปรแกรมทางการศึกษา คณิตศาสตร์ สถิติ และธุรกิจ เพราะฉะนั้นท่านสามารถจะเลือกฝึกได้อย่างจุใจ พลาดไม่ได้สำหรับหนังสือ 2 เล่มนี้ ความหนา 162 หน้า พิมพ์ด้วยกระดาษปอนด์อย่างดี *เล่มละ 45 บาท*

รวมโปรแกรมสำหรับแอปเปิ้ล เล่ม 1

และนี่คือโปรแกรมใช้งานสำหรับนักเล่นไมโครคอมพิวเตอร์ที่ต้องการจะใช้งานเครื่อง **APPLE II** ให้คุ้มค่า 28 โปรแกรม ได้รับการบรรจุแน่นในหนังสือเล่มนี้ โดยแบ่งออกเป็น 6 ภาค ตามลักษณะใช้งาน เช่น ภาคเกี่ยวกับเวลา ธุรกิจและการลงทุน การศึกษา เป็นต้น สำหรับโปรแกรมมันส์ ๆ ที่ท่านต้องการชื่นชอบ อาทิเช่น : -

- เครื่องควบคุมการปิดเปิดไฟฟ้าสารพัดประโยชน์
- เครื่องตั้งเวลาอัตโนมัติ - วิเคราะห์การใช้ไฟฟ้า
- การคิดค่าเสื่อมราคา - และอื่น ๆ อีกจุใจ...

ความหนา 152 หน้า พิมพ์ด้วยกระดาษปอนด์อย่างดี *ราคาเล่มละ 48 บาท*

รวมโปรแกรมสำหรับแอปเปิ้ล เล่ม 2

เป็นหนังสือที่ต่อเนื่องมาจากรวมโปรแกรมสำหรับแอปเปิ้ล เล่ม 1 โดยเล่มนี้มีโปรแกรมถึง 37 โปรแกรม เมื่อท่านนำไปใช้ท่านจะทึ่งกับความสามารถและความรอบรู้ของมัน เช่น : -

- โปรแกรมแบ่งเกรดของนักเรียน
- โปรแกรมค่าใช้จ่ายประจำเดือน
- โปรแกรมวิเคราะห์การใช้น้ำ
- โปรแกรมวิเคราะห์ค่าอาหาร
- โปรแกรมแนะนำการซื้อหุ้น
- ฯลฯ

ความหนา 228 หน้า พิมพ์ด้วยกระดาษปอนด์อย่างดี *ราคาเล่มละ 55 บาท*

นอกจากนี้เรายังมีหนังสือคอมพิวเตอร์อื่น ๆ อีกมากมายให้ท่านเลือกชมได้ เช่น : -

ภาษาเบสิค

ภาษาโฟแทรน 4

ไมโครโปรเซสเซอร์-ไมโครคอมพิวเตอร์

วารสารไมโครคอมพิวเตอร์

*สมาชิกวารสารของซีเอ็ดทุกฉบับสั่งซื้อได้ในราคาลด 15% ค่าจัดส่งฟรี !!

หาซื้อและเลือกชมได้ที่ สยามสแควร์ สนามหลวง วังบูรพา ร้านเมฆรุ่งเรือง ร้านเพ็ญบุญ ร้านดวงกมลสมัย

สั่งซื้อทางไปรษณีย์ ธนาณัติสั่งจ่าย ปท.ราชเทวีในนาม บริษัท ซีเอ็ดยูเคชั่น จำกัด 800/43-45 ซ.ตระกูลสุข ถ. อโศกดินแดง ห้วยขวาง กท.10400 โทร. 2450390

ค่าจัดส่งสำหรับผู้ไม่ได้เป็นสมาชิก

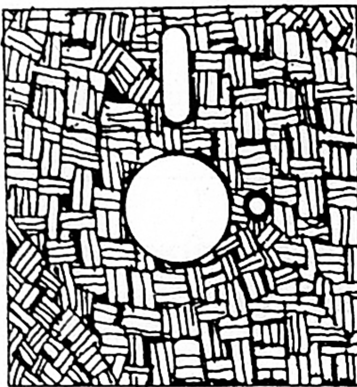
ซื้อไม่เกิน 100 บาท ค่าจัดส่ง 7 บาท ซื้อเกิน 100 บาท ค่าจัดส่ง 7%

พิมพ์เสียเหนื่อย กว่าจะหมดทุกโปรแกรม



หนังสือ *เรียน/เล่น/ใช้โปรแกรม APPLESOFT BASIC-ด้วยตนเอง* และหนังสือ *เรียน APPLESOFT BASIC ด้วยตนเอง เล่ม 2* ทั้งสองเล่มนี้ มีโปรแกรมประกอบการสอนตลอดเล่มอยู่มากมาย ซึ่งถ้าท่านจะคีย์ลงเครื่องทั้งหมดก็คงใช้เวลาไม่น้อย แต่ถ้าจะไม่คีย์เก็บไว้ในดิสค์ เวลาจะเรียกมาใช้ก็จะเสียเวลามาคีย์ใหม่มาก เราพอจะเล็งเห็นถึงปัญหานี้ เราจึง : -

ขอเสนอดิสค์โปรแกรมสำเร็จรูป





ทุกโปรแกรมซึ่งได้รับการบรรจุไว้แล้วในดิสค์ 2 หน้า และได้จัดทำ Menu เพื่อให้คุณสามารถเลือกโปรแกรมที่ต้องการได้ทันที เหมาะกับ : -

- *ผู้ที่มีสายตาไม่ดีและอายุ-มากแล้วแต่อยากเล่นกับโปรแกรม*
- *วัยรุ่นใจร้อนไม่อยากคีย์-โปรแกรมเอง แต่อยากเล่นโปรแกรม*
- *ผู้ที่ต้องการความรวดเร็วในการใช้งาน*

เราบริการดิสค์นี้ในราคาเพียงแผ่นละ **200** บาทเท่านั้น สั่งซื้อได้ที่ บริษัท ซีเอ็ดยูเคชั่น จำกัด เลขที่ 800/43-45 ซ.ตระกูลสุข ถ.อโศกดินแดง ห้วยขวาง กทม. 10400 โทร. 245-0390, 245-9216 ถ้าคุณอยู่ต่างจังหวัดส่งเช็ค/ธนาณัติ มายัง ปท.ดินแดง พร้อมค่าส่ง 7 บาท

หนังสือสำหรับผู้ศึกษา/ใช้/เล่นคอมพิวเตอร์



โปรแกรมคอมพิวเตอร์ภาษาเบสิค

โดย ยืน ภู่วรวรรณ พิชิต สุขเจริญพงศ์ สมนึก คีรีโต และภิญโญ ศรีสุขินวงศ์ ราคา 55 บาท

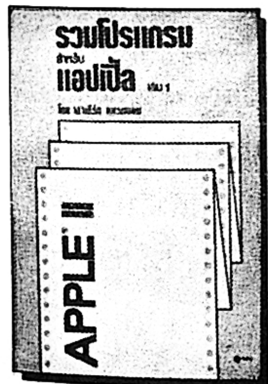
เป็นหนังสือภาษาเบสิคอีกเล่มหนึ่งที่จัดทำขึ้น โดยคณะอาจารย์จากห้องปฏิบัติการวิจัยไมโครคอมพิวเตอร์ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตร โดยพยายามสร้างพื้นฐานการเขียนโปรแกรมสำหรับผู้เริ่มต้น การประยุกต์ใช้งานของภาษาเบสิคในเรื่องการเขียนภาพ การสร้างเกมการใช้งานทางคณิตศาสตร์และสถิติ การใช้งานทางด้านธุรกิจ การบริหาร



คอมพิวเตอร์กับการเขียนโปรแกรมภาษาเบสิค

โดย พรทิพย์ เลาหวิโรจน์ และสุพจน์ จิตต์ประเสริฐ ราคา 65 บาท

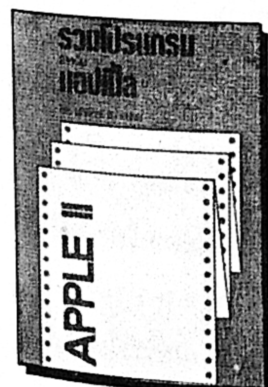
หนังสือเล่มนี้สำหรับผู้ที่ต้องการเรียนและเขียนภาษา APPLESOFT BASIC ด้วยตนเองให้เป็นภายใน 50 ชั่วโมง โดยจะเริ่มต้นตั้งแต่พื้นฐานของการโปรแกรมด้วยภาษาเบสิค อ้างอิงถึงการใช้งานกับเครื่องยี่ห้อ APPLE โดยจะค่อย ๆ พัฒนาคุณขึ้นมาจนถึงระดับที่จะโปรแกรมงานบางอย่างได้เอง



รวมโปรแกรมสำหรับแอปเปิ้ล เล่ม 1

โดย เฮาเอิร์ด เบเรนบอน ราคา 48 บาท

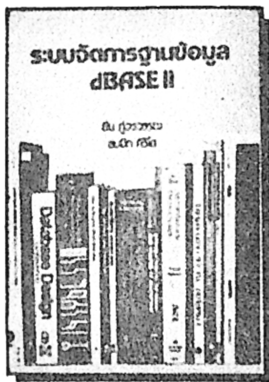
และนี่คือ โปรแกรมใช้งานสำหรับนักเล่นไมโครคอมพิวเตอร์ที่ต้องการจะใช้เครื่อง APPLE II ให้คุ้มค่า มี 28 โปรแกรมอยู่ภายในเล่มแบ่งเป็น 6 ภาคตามลักษณะการใช้งานเช่น ภาคเกี่ยวกับเวลา ธุรกิจและการลงทุน การศึกษา เป็นต้น ตัวอย่างโปรแกรม ได้แก่เครื่องควบคุมการปิดเปิดไฟฟ้าสารพัดประโยชน์ เครื่องตั้งเวลาอัตโนมัติ และคิดค่าเสื่อมราคาวิเคราะห์การใช้ไฟฟ้า หมอดูไพ่ป๊อก และเกมฝึกสมองอีกมากมาย



รวมโปรแกรมสำหรับแอปเปิ้ล เล่ม 2

โดย เฮาเอิร์ด เบเรนบอน ราคา 55 บาท

เป็นหนังสือรวมโปรแกรมใช้งานเครื่อง APPLE II อีก 37 โปรแกรมอาทิเช่น การคิดดอกเบี้ยทบต้นโปรแกรมเก็บหมายเลขโทรศัพท์ โปรแกรมรักษาสภาพสมดุลของบัญชีเช็ค โปรแกรมแนะนำการซื้อหุ้นทดสอบความสามารถทางจิต โปรแกรมแบ่งเกรดนักเรียนและโปรแกรมทดสอบความสามารถอีกมากมาย



ระบบจัดการฐานข้อมูล dBASE II

โดย ยืน ภู่วรวรรณ และสมนึก คีรีโต ราคา 60 บาท

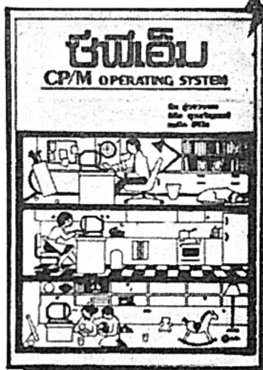
เป็นหนังสือสำหรับผู้ที่ยังไม่เคยใช้ไมโครคอมพิวเตอร์ในระบบจัดการฐานข้อมูลมาก่อนเลย และสนใจการใช้งาน dBASE II ซึ่งเป็นโปรแกรมที่มีประสิทธิภาพสูงสุดและได้รับความนิยมมากโปรแกรมหนึ่ง ภายในเล่มแบ่งเป็น 3 ตอน ตอนแรกจะวางพื้นฐานของระบบจัดการฐานข้อมูลตลอดจนโปรแกรมโครงสร้าง ตอนที่สองว่าด้วยวิธีการทำงานของ dBASE II และตอนที่สามว่าด้วยการนำเอา dBASE II มาใช้งาน



อิเล็กทรอนิกส์เวอร์คชีต

โดย ยืน ภู่วรวรรณ และอัศนีย์ ก่อตระกูล ราคา 59 บาท

เหมาะสำหรับผู้ที่เรียนรู้การใช้งานคอมพิวเตอร์มาบ้างแล้ว และกำลังต้องการนำไมโครคอมพิวเตอร์มาใช้งานจริงโดยจะให้หลักการและวิธีใช้อิเล็กทรอนิกส์เวอร์คชีต ซึ่งใช้ได้ทั้งวิสิแคลค์ มัลติแพลน ซูเปอร์แคลค์ (ทั้งกำลัง 2 และกำลัง 3) และโลตัส 123 ตัวอย่างของอิเล็กทรอนิกส์เวอร์คชีตในหนังสือเล่มนี้ใช้กับโปรแกรมซูเปอร์แคลค์เป็นหลัก แต่ก็มีหลักการใช้หรือการเรียกคล้ายอิเล็กทรอนิกส์เวอร์คชีตอีกหลายตัว



ซีพีเอ็ม (CP/M Operating System)

โดย ยืน ภู่วรวรรณ พิชิต สุขเจริญพงษ์ และสมนึก คีรีโต ราคา 58 บาท

ภายในหนังสือเล่มนี้ได้ให้รายละเอียดเกี่ยวกับโครงสร้างภายในของซีพีเอ็มการเชื่อมต่อทางซอฟต์แวร์เพื่อพัฒนาโปรแกรมหรือดัดแปลงแก้ไขโปรแกรมการจัดการระบบข้อมูลบนแผ่นดิสเก็ตต์ รายละเอียดเกี่ยวกับคำสั่งต่าง ๆ ที่ใช้ในซีพีเอ็ม ทั้งเป็นคำสั่งที่อยู่ภายในตัวซีพีเอ็มเองและคำสั่งประกอบที่อยู่ภายนอก ทำให้ผู้ศึกษาสามารถเรียนรู้และเข้าใจเกี่ยวกับการพัฒนา หรือเขียนโปรแกรมภาษาแอสเซมบลี้ได้เป็นอย่างดี



โปรแกรมคอมพิวเตอร์ภาษาฟอร์แทรน

โดย ยืน ภู่วรวรรณ พิชิต สุขเจริญวงศ์ และนนทวัฒน์ จันทร์เจริญ ราคา 68 บาท

หนังสือเล่มนี้เหมาะสำหรับใช้เป็นตำราในระดับวิทยาลัย มหาวิทยาลัย ตลอดจนผู้ที่จะศึกษาด้วยตนเอง เนื้อหาภายในได้วางโครงร่างในลักษณะต่อเรียงกันไปโดยเริ่มจากหลักการทางคอมพิวเตอร์ สิ่งที่ควรรู้เกี่ยวกับคอมพิวเตอร์ หลักการทางด้านภาษาคอมพิวเตอร์ การเขียนแผนภูมิสายงานหรือผังงานการใช้คำสั่ง

ต่าง ๆ ที่มีในภาษาฟอร์แทรนและตอนสุดท้ายว่าด้วยการประยุกต์ภาษาฟอร์แทรนมาใช้งาน



โปรแกรมคอมพิวเตอร์เบื้องต้น ภาษาโฟแทรน 4

โดย เดือน สินธุพันธ์ประทุม ราคา 49 บาท

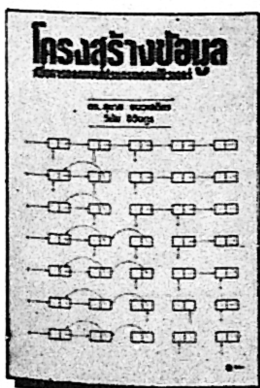
เป็นหนังสืออีกเล่มหนึ่งที่ได้รับความสนใจอย่างสูงจากผู้อ่านจนต้องพิมพ์เป็นครั้งที่ 4 เนื่องจากเนื้อหาสามารถอ่านเข้าใจได้ง่ายทั้งผู้เริ่มเรียนหรือผู้ที่พอมีพื้นความรู้มาบ้างแล้วคำสั่งพื้นฐานจากหนังสือเล่มนี้สามารถใช้ได้กับคอมพิวเตอร์ทุกเครื่องที่รับภาษาโฟแทรน 4 ได้ และยังเพิ่มเติมคำสั่งบางประเภทซึ่งใช้ได้เฉพาะเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพสูงด้วยในบทที่ 10



โปรแกรมคอมพิวเตอร์ ภาษา RPG

โดยวิชิต ปุณวัตร์ ราคา 140 บาท

ภาษา RPG มีลักษณะที่เหมาะสมกับงานธุรกิจเป็นส่วนใหญ่ ทั้งนี้เพราะสามารถเขียนรายงานแบบต่าง ๆ ทางธุรกิจได้ง่ายและรวดเร็ว แต่จุดอ่อนของภาษานี้อยู่ที่ไม่สามารถเขียน Program Flow ได้อย่างอิสระ ทำให้การใช้ภาษานี้ยังอยู่ในวงจำกัด อย่างไรก็ดีภาษานี้ยังมีคุณลักษณะที่ดีเด่นอีกหลายประการ ภาษา RPG จึงควรได้รับการปรับปรุงแก้ไขโดยการนำเอาเทคนิคการเขียนโปรแกรมโครงสร้าง (Structured Programming) มาประยุกต์ใช้ทำให้ RPG สามารถเขียน Program Flow ได้และทั้งภาษา RPG ได้ถูกปรับปรุงเป็น RGP II และ RPG III ตามลำดับ ทำให้ขีดความสามารถของภาษาสูงขึ้นเป็นอย่างมากจนกล่าวว่าถ้ารู้จักใช้แล้วจะติดใจ



โครงสร้างข้อมูล

โดย สุชาย ธนวเสถียร และวิชัย จิวังกูร ราคา 54 บาท

หลังจากที่เรียนภาษาคอมพิวเตอร์ตลอดจนการเขียนโปรแกรมแล้ว ขั้นต่อไปคือเรียนวิชาโครงสร้างข้อมูล วิชานี้มีความสำคัญอย่างมากเนื่องจากเป็นวิชาการพื้นฐานทางคอมพิวเตอร์ที่จะช่วยให้ผู้เขียนโปรแกรมทุกคนสามารถใช้โครงสร้างข้อมูลแบบที่เหมาะสม ซึ่งช่วยให้โปรแกรมทำงานเร็วขึ้นกินเนื้อที่น้อยลง และมีประสิทธิภาพตามที่ต้องการ ฉะนั้นถ้าท่านได้เริ่มเรียนรู้การใช้คอมพิวเตอร์โดยใช้เครื่องไมโครไม่ว่าจะเป็น APPLE จริงหรือเทียมหรือเครื่องยี่ห้อใดก็ตาม หนังสือเล่มนี้จะเป็นเล่มที่คุณพลาดไม่ได้



ไมโครโปรเซสเซอร์-ไมโครคอมพิวเตอร์

โดย ยืน ภู่วรวรรณ และวัฒนา เชียงกูล ราคา 75 บาท

หนังสือเล่มนี้เป็นการวางพื้นฐานของไมโครคอมพิวเตอร์ โดยเน้นให้เห็นถึงโครงสร้างคอมพิวเตอร์ที่จำเป็นต้องมีซีพียูหน่วยความจำ และอินพุท–เอาท์พุท นอกจากนี้ยังกล่าวถึงรายละเอียดของซีพียู Z-80 และการต่อเป็นระบบไมโครคอมพิวเตอร์ การเขียนโปรแกรม การประยุกต์ใช้งาน และสุดท้ายจะเป็นบทที่ว่าด้วยอุปกรณ์สนับสนุน ระบบไมโครคอมพิวเตอร์ที่สำคัญซึ่งได้แก่ 8214, 8212, Z-80 PIO, Z-80 CTC และ 8251



เรื่องน่ารู้เกี่ยวกับไมโครคอมพิวเตอร์

โดย ยืน ภู่วรวรรณ ราคา 53 บาท

เหมาะสำหรับผู้ที่อยากรู้รายละเอียดทั่วไปเกี่ยวกับไมโครคอมพิวเตอร์ เพื่อศึกษาข้อมูลรายละเอียดต่าง ๆ โดยเนื้อหาจะเน้นในลักษณะเรื่องทั่วไปที่เกี่ยวข้องถึงฮาร์ดแวร์ ซึ่งได้แก่ส่วนประกอบต่าง ๆ และส่วนของซอฟท์แวร์ที่ใช้กับไมโครคอมพิวเตอร์ จึงเปรียบเสมือนคู่มือที่ผู้สนใจในระดับพื้นฐานควรมีไว้เพราะสามารถศึกษารายละเอียดต่าง ๆ รวมทั้งเป็นข้อมูลสำหรับการจัดซื้อการพิจารณาสเปค และสร้างฐานความรู้เพื่อศึกษาไมโครคอมพิวเตอร์ในระดับที่ลึกซึ้งขึ้นไป



เรียน APPLESOFT BASIC ด้วยตนเอง (เล่ม 1)

โดย พ.ต.ประพัฒน์ อุทโยภาศ ราคา 45 บาท

หนังสือเล่มนี้จะช่วยให้ท่านสามารถเขียนโปรแกรมสั่งงานไมโครคอมพิวเตอร์ได้ด้วยตัวท่านเอง โดยจะเริ่มตั้งแต่คอมพิวเตอร์คืออะไร เป็นอย่างไร มีอะไรบ้าง จะต้องสั่งงานอย่างไร คำสั่งที่ใช้มีอะไรบ้าง ตลอดเล่มจะมีโปรแกรมที่ทำการทดสอบแล้วโดยผ่านเครื่อง APPLE II ให้ท่านฝึกฝนโดยเริ่มจากง่าย ๆ ขึ้นไป จนในที่สุดท่านก็จะสามารถอ่านโปรแกรมเข้าใจได้ด้วยตัวท่านเอง



เรียน APPLESOFT BASIC ด้วยตนเอง (เล่ม 2)

โดย พ.ต.ประพัฒน์ อุทโยภาศ ราคา 42 บาท

เป็นหนังสือเพิ่มความรู้สำหรับผู้ที่อ่านเล่ม 1 มาแล้ว เพื่อจะได้ศึกษาส่วนที่ผู้เขียนเว้นไว้เพื่อมิให้สับสนในระยะที่เริ่มเรียนรู้ บทต่าง ๆ ภายในเล่มมีลึกเข้าไปในคอมพิวเตอร์, เพิ่มเติมเสริมแต่ง, FUNCTIONS, กราฟิก, SHAPE TABLE, TEXT FILE และ FLOW CHART



เรียน/เล่น/ใช้ โปรแกรม APPLESOFT BASIC

โดย พ.ต.ประพัฒน์ อุทโยภาศ ราคา 45 บาท

เมื่อท่านสามารถอ่านโปรแกรมเข้าใจแล้ว หนังสือเล่มนี้จะเป็นเล่มต่อไปที่ช่วยเพิ่มประสบการณ์แก่ท่านในด้านการหัดป้อนข้อมูล (KEY) หัด SAVE, LOAD, LIST และ RUN หัดแก้และหัดแปลงโปรแกรม ซึ่งต่อไปก็จะสามารถเขียนโปรแกรมได้เอง สำหรับโปรแกรม 40 โปรแกรมที่ได้รวบรวมไว้ในเล่มมีทั้งที่เป็นเกม โปรแกรมทางการศึกษา คณิตศาสตร์ สถิติ และธุรกิจ เพราะฉะนั้นท่านสามารถจะเลือกฝึกได้อย่างจุใจ





เรียนมัลติเพลนด้วยตนเอง

โดย ประพัฒน์ อุทโยภาศ ราคา 46 บาท

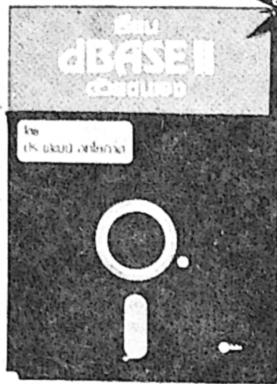
หากท่านมีนิ้วมือเหลืออย่างน้อย 2 นิ้ว พออ่านภาษาอังกฤษออก มีไมโครคอมพิวเตอร์ที่มีหน่วยความจำอย่างน้อย 64 K BYTE มีโปรแกรมสำเร็จรูปชื่อมัลติเพลน และได้อ่านหนังสือเล่มนี้เพียงครึ่งเล่ม ท่านก็สามารถนำคอมพิวเตอร์ไปใช้ประโยชน์ได้อย่างกว้างขวาง อาทิเช่น ทำบัญชี สินค้าคงคลัง ออกอินวอยซ์ เก็บหมายเลขโทรศัพท์ ฯลฯ หนังสือเล่มนี้ใช้ภาษาง่าย ๆ อย่างเป็นกันเองเหมือนเพื่อนคุยกัน แต่ขณะเดียวกันจะให้ความรู้ที่แม้แต่ในหนังสือคู่มือการใช้โปรแกรมนี้เองก็ไม่ได้กล่าวไว้ กล่าวได้ว่าหนังสือเล่มนี้จะเป็นประโยชน์แก่ทุกคน ตั้งแต่ผู้เริ่มใช้ จนกระทั่งผู้ที่ได้เคยใช้โปรแกรมนี้มาอย่างช่ำชอง



เทคนิคการใช้งานเครื่องพิมพ์ EPSON ด้วย APPLESOFT BASIC

โดย พ.ต.ประพัฒน์ อุทโยภาศ ราคา 22 บาท

หนังสือเล่มนี้จะบอกวิธีใช้เครื่องพิมพ์ EPSON ที่พ่วงกับเครื่องคอมพิวเตอร์ตระกูล APPLE มีการสร้างตัวหนังสือแบบต่าง ๆ ถึง 11 แบบ การปรับระยะบรรทัด การพิมพ์ภาพกราฟิกพร้อมกับมีโปรแกรมแถมให้ การพิมพ์นามบัตรสำเร็จรูป การพิมพ์กราฟแท่ง มีตารางสรุปคำสั่งต่าง ๆ อย่างที่จะหาไม่ได้ในหนังสือทั้งไทยและเทศ และท้ายเล่มมีโปรแกรมสั้น ๆ สนุก ๆ สำหรับเด็ก ๆ หนังสือเล่มนี้เหมาะสำหรับผู้ที่มีเครื่องพิมพ์ EPSON อยู่แล้วหรือผู้ที่กำลังคิดจะมี





เรียน dBASE II ด้วยตนเอง

โดย ประพัฒน์ อุทโยภาศ ราคา 47 บาท

เรียน dBASE II ด้วยตนเองเล่มนี้ เป็นหนังสือคอมพิวเตอร์ที่ผู้อ่านสามารถจะเรียนรู้ได้โดยไม่จำเป็นจะต้องมีไมโครคอมพิวเตอร์อยู่ใกล้มือ เพราะผู้เขียนได้แจกแจงตามลำดับ โดยมีภาพที่จะปรากฏอยู่บนจอภาพจริง ๆ แสดงให้เห็นทุกขั้นตอน ผู้ที่ไม่มีพื้นฐานในเรื่องไมโครคอมพิวเตอร์สามารถติดตามและจะนำไปใช้ประโยชน์ได้ในที่สุด หนังสือเล่มนี้ใช้ได้กับไมโครคอมพิวเตอร์ทุกชนิดที่สามารถใช้ซอฟต์แวร์ dBASE II ได้

หาซื้อหรือเลือกชมได้ที่ : ร้านหนังสือย่านสยามสแควร์ วังบูรพา สนามหลวง ศูนย์หนังสือจุฬาฯ ดวงกมลสมัย (ข้างโรงภาพยนตร์ O.A) เพ็ญบุญ (สะพานควาย) ชมอักษร (ชั้น 2 เซ็นทรัลพลาซ่า ลาดพร้าว) เมฆรุ่งเรือง (สามแยก ม.เกษตรศาสตร์) สหกรณ์มหาวิทยาลัยเกษตรศาสตร์ BOOK WORM (สุขุมวิทระหว่างซอย 53-55) และที่ซีเอ็ด (อโศก-ดินแดง)

ต่างจังหวัดซื้อได้ที่ : สุริวงศ์บุ๊คเซ็นเตอร์ (เชียงใหม่) เจริญรัตน์ (ลำปาง) เสียงทิพย์บุ๊คเซ็นเตอร์ (พิษณุโลก) สิทธา (พิษณุโลก) ศรีไกรลาศ (นครสวรรค์) นิยมศึกษา (ลพบุรี) ศรีเมืองบุ๊คเซ็นเตอร์ (ร้อยเอ็ด) ฮอลลีวู้ดบุ๊ค (อุบลฯ) ไทยวารี (ขอนแก่น) โพธิ์ทองบุ๊คเซ็นเตอร์ (นครราชสีมา) บุ๊คเซ็นเตอร์ (นครราชสีมา) เล้งโห (ภูเก็ต) วงเวียนบุ๊คเซ็นเตอร์ (นครศรีธรรมราช) มิตรสาส์น (ตรัง) นิรมล (สงขลา) บรรณาคาร (หาดใหญ่)

หรือสั่งซื้อทางไปรษณีย์ได้ที่ : บริษัท ซีเอ็ดยูเคชั่น จำกัด
800/43-45 ซอยตระกูลสุข ถนนอโศก-ดินแดง ห้วยขวาง กทม. 10400 ธนาณัติสั่งจ่าย ปท.ดินแดง ในนามบริษัทฯ

ค่าจัดส่ง ซื้อไม่เกิน 100 บาท ค่าจัดส่ง 7 บาท ซื้อเกิน 100 บาท ค่าจัดส่ง 7%

พิเศษ เฉพาะสมาชิกวารสารของซีเอ็ดทุกฉบับ สั่งซื้อได้ในราคาลด 15% ค่าส่งฟรี โดยเพียงระบุหมายเลขสมาชิกมาเท่านั้น



บริษัท ซีเอ็ดยูเคชั่น จำกัด

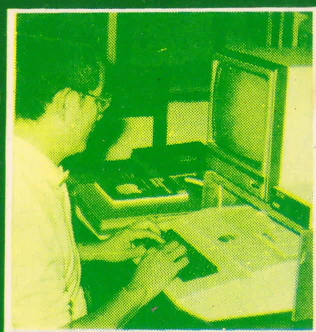
ผู้ผลิตวารสารและหนังสือวิชาการด้านวิทยาศาสตร์และเทคโนโลยี



เรียน APPLESOFT BASIC ด้วยตนเอง เล่ม 2

หนังสือเล่มนี้มีวัตถุประสงค์ที่จะเพิ่มความรู้สำหรับผู้ที่ศึกษาเรื่อง ไมโครคอมพิวเตอร์มาบ้าง โดยเฉพาะผู้ที่ได้อ่าน "เรียน APPLESOFT BASIC ด้วยตนเอง" มาแล้วควรที่จะอ่านหนังสือเล่มนี้ต่อ เพื่อจะได้ศึกษาส่วนที่ผู้เขียนเว้นไว้เพื่อมิให้สับสนในระยะที่เริ่มเรียนรู้ และนอกจากนี้แล้วผู้อ่านยังสามารถจะนำความรู้ไปทำการเขียนโปรแกรมง่าย ๆ ซึ่งใช้ประโยชน์ได้มาก จุดเด่นของหนังสือเล่มนี้อยู่ที่การใช้ภาษาง่าย ๆ ที่ทำให้คนไม่มีพื้นฐานก็สามารถเข้าใจได้

นอกจากหนังสือเล่มนี้แล้ว พ.ต.ประพัฒน์ ยังมีผลงานการเรียนอีกหลายชิ้น แต่ที่ลือลั่นในวงการไมโครคอมพิวเตอร์มากก็คือ หนังสือ "เรียน APPLESOFT BASIC ด้วยตนเอง" ซึ่งแม้จะเป็นผลงานด้านหนังสือเล่มแรก แต่ก็ทำสถิติจนทิ้งไปทั้งวงการที่สามารถจำหน่ายได้มากกว่าหนึ่งหมื่นเล่มภายในเวลาเพียงไม่กี่เดือน



พ.ต. ประพัฒน์ อุทโยภาศ จบปริญญาตรี และโท ทางวิศวกรรมศาสตร์ จากจุฬาลงกรณ์มหาวิทยาลัย สมัยที่ ณ เณร ยังไม่มีตัวการันต์ ในสาขาวิชาที่ยังไม่เกี่ยวแก่คอมพิวเตอร์ เพราะสมัยนั้นยังไม่มีการสอนกัน จบออกมาก็ทำงานอยู่กรมโยธา ซึ่งสมัยนั้นมีห้อยท้ายว่า "เทศบาล" แทนที่จะเป็น "ธิการ" ทำให้สับสนกับเทศบาลนครกรุงเทพบ่อย ๆ อยู่มาได้ 3-4 ปีก็ขอโอนไปสอนหนังสือที่โรงเรียนนายร้อยพระจุลจอมเกล้า อยู่ได้ถึง 10 ปี จนได้รับพระราชทานยศเป็นพันตรี บริษัทฯ ฝรั่งเสนอจ้างในอัตราเงินเดือนที่สูงกว่าจอมพลจึงทนเย้ายวนใจไม่ไหว อยู่ได้ 9 ปีกว่า ๆ เงินเดือนชักเยอะแต่เขี้ยวชักจะคลอนเลยต้องออกมา เดี๋ยวนี้เป็นรอง-ผู้จัดการบริษัทกรุงเทพเอ็นยิเนียริ่งคอนซัลแตนท์ จำกัด ตั้งใจว่าจะอยู่ไปจนเกษียณ ถ้าไม่ถูกเขาไล่ออกเพราะหมกมุ่นกับ APPLE มากเกินไป

ISBN 974-7965-03-8